Part-1-Data-Loading

December 21, 2021

1 Load Data

In this File we load the raw data and convert it into JSON format.

Whilst the data was said to be in JSON format, when running it through a JSON linter, we noticed that it was not proper JSON due to the use of single quotes instead of double quotes.

To remedy this, we explored options such as replacing the single quotes with double quotes. However this led to issues where the name of a game included an apostrophe, e.g. `Assassin's Creed`.

1.1 User-items data

The dataset was downloaded from https://cseweb.ucsd.edu/~jmcauley/datasets.html#steam_data file *Version 1: User and Item Data*.

[ ]:
```
import pandas as pd
import numpy as np
import ast
import json
```

[ ]:
```
with open("C:
↪\\Users\\pmogh\\Documents\\Python_Workspace\\Final_Project_Dataset\\ProjectData\\Part_1\\us
↪json",encoding="utf-8") as player_item_file:
    lines = player_item_file.readlines()[0:10000]
```

[ ]:
```
lines[0]
```

[ ]:
```
'{\'user_id\': \'76561197970982479\', \'items_count\': 277, \'steam_id\':
\'76561197970982479\', \'user_url\':
\'http://steamcommunity.com/profiles/76561197970982479\', \'items\':
[{\'item_id\': \'10\', \'item_name\': \'Counter-Strike\', \'playtime_forever\':
6, \'playtime_2weeks\': 0}, {\'item_id\': \'20\', \'item_name\': \'Team Fortress
Classic\', \'playtime_forever\': 0, \'playtime_2weeks\': 0}, {\'item_id\':
\'30\', \'item_name\': \'Day of Defeat\', \'playtime_forever\': 7,
\'playtime_2weeks\': 0}, {\'item_id\': \'40\', \'item_name\': \'Deathmatch
Classic\', \'playtime_forever\': 0, \'playtime_2weeks\': 0}, {\'item_id\':
\'50\', \'item_name\': \'Half-Life: Opposing Force\', \'playtime_forever\': 0,
\'playtime_2weeks\': 0}, {\'item_id\': \'60\', \'item_name\': \'Ricochet\',
\'playtime_forever\': 0, \'playtime_2weeks\': 0}, {\'item_id\': \'70\',
```

\'item_name\': \'Half-Life\', \'playtime_forever\': 0, \'playtime_2weeks\': 0},
{\'item_id\': \'130\', \'item_name\': \'Half-Life: Blue Shift\',
\'playtime_forever\': 0, \'playtime_2weeks\': 0}, {\'item_id\': \'300\',
\'item_name\': \'Day of Defeat: Source\', \'playtime_forever\': 4733,
\'playtime_2weeks\': 0}, {\'item_id\': \'240\', \'item_name\': \'Counter-Strike:
Source\', \'playtime_forever\': 1853, \'playtime_2weeks\': 0}, {\'item_id\':
\'3830\', \'item_name\': \'Psychonauts\', \'playtime_forever\': 333,
\'playtime_2weeks\': 0}, {\'item_id\': \'2630\', \'item_name\': \'Call of Duty
2\', \'playtime_forever\': 75, \'playtime_2weeks\': 0}, {\'item_id\': \'3900\',
\'item_name\': "Sid Meier\'s Civilization IV", \'playtime_forever\': 338,
\'playtime_2weeks\': 0}, {\'item_id\': \'34440\', \'item_name\': "Sid Meier\'s
Civilization IV", \'playtime_forever\': 0, \'playtime_2weeks\': 0},
{\'item_id\': \'3920\', \'item_name\': "Sid Meier\'s Pirates!",
\'playtime_forever\': 2, \'playtime_2weeks\': 0}, {\'item_id\': \'6400\',
\'item_name\': \'Joint Task Force\', \'playtime_forever\': 286,
\'playtime_2weeks\': 0}, {\'item_id\': \'6910\', \'item_name\': \'Deus Ex: Game
of the Year Edition\', \'playtime_forever\': 2685, \'playtime_2weeks\': 0},
{\'item_id\': \'7670\', \'item_name\': \'BioShock\', \'playtime_forever\': 633,
\'playtime_2weeks\': 0}, {\'item_id\': \'409710\', \'item_name\': \'BioShock
Remastered\', \'playtime_forever\': 0, \'playtime_2weeks\': 0}, {\'item_id\':
\'220\', \'item_name\': \'Half-Life 2\', \'playtime_forever\': 696,
\'playtime_2weeks\': 0}, {\'item_id\': \'320\', \'item_name\': \'Half-Life 2:
Deathmatch\', \'playtime_forever\': 0, \'playtime_2weeks\': 0}, {\'item_id\':
\'340\', \'item_name\': \'Half-Life 2: Lost Coast\', \'playtime_forever\': 37,
\'playtime_2weeks\': 0}, {\'item_id\': \'360\', \'item_name\': \'Half-Life
Deathmatch: Source\', \'playtime_forever\': 0, \'playtime_2weeks\': 0},
{\'item_id\': \'380\', \'item_name\': \'Half-Life 2: Episode One\',
\'playtime_forever\': 168, \'playtime_2weeks\': 0}, {\'item_id\': \'400\',
\'item_name\': \'Portal\', \'playtime_forever\': 173, \'playtime_2weeks\': 0},
{\'item_id\': \'420\', \'item_name\': \'Half-Life 2: Episode Two\',
\'playtime_forever\': 323, \'playtime_2weeks\': 0}, {\'item_id\': \'9340\',
\'item_name\': \'Company of Heroes: Opposing Fronts\', \'playtime_forever\':
692, \'playtime_2weeks\': 0}, {\'item_id\': \'228200\', \'item_name\': \'Company
of Heroes (New Steam Version)\', \'playtime_forever\': 0, \'playtime_2weeks\':
0}, {\'item_id\': \'11450\', \'item_name\': \'Overlord\', \'playtime_forever\':
0, \'playtime_2weeks\': 0}, {\'item_id\': \'7940\', \'item_name\': \'Call of
Duty 4: Modern Warfare\', \'playtime_forever\': 1185, \'playtime_2weeks\': 0},
{\'item_id\': \'4700\', \'item_name\': \'Medieval II: Total War\',
\'playtime_forever\': 477, \'playtime_2weeks\': 0}, {\'item_id\': \'12900\',
\'item_name\': \'Audiosurf\', \'playtime_forever\': 115, \'playtime_2weeks\':
0}, {\'item_id\': \'13250\', \'item_name\': \'Unreal Gold\',
\'playtime_forever\': 0, \'playtime_2weeks\': 0}, {\'item_id\': \'16100\',
\'item_name\': \'Virtual Villagers: A New Home\', \'playtime_forever\': 0,
\'playtime_2weeks\': 0}, {\'item_id\': \'15700\', \'item_name\': "Oddworld:
Abe\'s Oddysee", \'playtime_forever\': 28, \'playtime_2weeks\': 0},
{\'item_id\': \'15710\', \'item_name\': "Oddworld: Abe\'s Exoddus",
\'playtime_forever\': 0, \'playtime_2weeks\': 0}, {\'item_id\': \'17330\',

\'item_name\': \'Crysis Warhead\', \'playtime_forever\': 31,
\'playtime_2weeks\': 0}, {\'item_id\': \'17340\', \'item_name\': \'Crysis
Wars\', \'playtime_forever\': 0, \'playtime_2weeks\': 0}, {\'item_id\':
\'22000\', \'item_name\': \'World of Goo\', \'playtime_forever\': 195,
\'playtime_2weeks\': 0}, {\'item_id\': \'500\', \'item_name\': \'Left 4 Dead\',
\'playtime_forever\': 513, \'playtime_2weeks\': 0}, {\'item_id\': \'4560\',
\'item_name\': \'Company of Heroes\', \'playtime_forever\': 1061,
\'playtime_2weeks\': 0}, {\'item_id\': \'17460\', \'item_name\': \'Mass
Effect\', \'playtime_forever\': 1613, \'playtime_2weeks\': 0}, {\'item_id\':
\'10500\', \'item_name\': \'Empire: Total War\', \'playtime_forever\': 186,
\'playtime_2weeks\': 0}, {\'item_id\': \'24740\', \'item_name\': \'Burnout
Paradise: The Ultimate Box\', \'playtime_forever\': 0, \'playtime_2weeks\': 0},
{\'item_id\': \'22200\', \'item_name\': \'Zeno Clash\', \'playtime_forever\':
271, \'playtime_2weeks\': 0}, {\'item_id\': \'26800\', \'item_name\': \'Braid\',
\'playtime_forever\': 445, \'playtime_2weeks\': 0}, {\'item_id\': \'1250\',
\'item_name\': \'Killing Floor\', \'playtime_forever\': 10006,
\'playtime_2weeks\': 0}, {\'item_id\': \'35420\', \'item_name\': \'Killing Floor
Mod: Defence Alliance 2\', \'playtime_forever\': 0, \'playtime_2weeks\': 0},
{\'item_id\': \'3590\', \'item_name\': \'Plants vs. Zombies: Game of the Year\',
\'playtime_forever\': 4413, \'playtime_2weeks\': 0}, {\'item_id\': \'8880\',
\'item_name\': \'Freedom Force\', \'playtime_forever\': 0, \'playtime_2weeks\':
0}, {\'item_id\': \'8890\', \'item_name\': \'Freedom Force vs. the 3rd Reich\',
\'playtime_forever\': 0, \'playtime_2weeks\': 0}, {\'item_id\': \'23120\',
\'item_name\': \'Droplitz\', \'playtime_forever\': 53, \'playtime_2weeks\': 0},
{\'item_id\': \'35700\', \'item_name\': \'Trine\', \'playtime_forever\': 199,
\'playtime_2weeks\': 0}, {\'item_id\': \'10140\', \'item_name\': \'3D Ultra
Minigolf Adventures Deluxe\', \'playtime_forever\': 25, \'playtime_2weeks\': 0},
{\'item_id\': \'35010\', \'item_name\': \'Batman: Arkham Asylum\',
\'playtime_forever\': 570, \'playtime_2weeks\': 0}, {\'item_id\': \'35140\',
\'item_name\': \'Batman: Arkham Asylum GOTY Edition\', \'playtime_forever\': 0,
\'playtime_2weeks\': 0}, {\'item_id\': \'29180\', \'item_name\': \'Osmos\',
\'playtime_forever\': 940, \'playtime_2weeks\': 0}, {\'item_id\': \'15520\',
\'item_name\': \'AaAaAA!!! - A Reckless Disregard for Gravity\',
\'playtime_forever\': 110, \'playtime_2weeks\': 0}, {\'item_id\': \'32370\',
\'item_name\': \'STAR WARS : Knights of the Old Republic \',
\'playtime_forever\': 5, \'playtime_2weeks\': 0}, {\'item_id\': \'37700\',
\'item_name\': \'Darkest of Days\', \'playtime_forever\': 782,
\'playtime_2weeks\': 0}, {\'item_id\': \'6020\', \'item_name\': \'STAR WARS
Jedi Knight: Jedi Academy \', \'playtime_forever\': 77, \'playtime_2weeks\': 0},
{\'item_id\': \'24860\', \'item_name\': \'Battlefield 2\', \'playtime_forever\':
437, \'playtime_2weeks\': 0}, {\'item_id\': \'39530\', \'item_name\':
\'Painkiller: Black Edition\', \'playtime_forever\': 503, \'playtime_2weeks\':
0}, {\'item_id\': \'550\', \'item_name\': \'Left 4 Dead 2\',
\'playtime_forever\': 1474, \'playtime_2weeks\': 0}, {\'item_id\': \'223530\',
\'item_name\': \'Left 4 Dead 2 Beta\', \'playtime_forever\': 0,
\'playtime_2weeks\': 0}, {\'item_id\': \'8980\', \'item_name\': \'Borderlands\',
\'playtime_forever\': 3061, \'playtime_2weeks\': 0}, {\'item_id\': \'41500\',

\'item_name\': \'Torchlight\', \'playtime_forever\': 536, \'playtime_2weeks\': 0}, {\'item_id\': \'20900\', \'item_name\': \'The Witcher: Enhanced Edition\', \'playtime_forever\': 139, \'playtime_2weeks\': 0}, {\'item_id\': \'10180\', \'item_name\': \'Call of Duty: Modern Warfare 2\', \'playtime_forever\': 1886, \'playtime_2weeks\': 0}, {\'item_id\': \'10190\', \'item_name\': \'Call of Duty: Modern Warfare 2 - Multiplayer\', \'playtime_forever\': 1784, \'playtime_2weeks\': 0}, {\'item_id\': \'17450\', \'item_name\': \'Dragon Age: Origins\', \'playtime_forever\': 4431, \'playtime_2weeks\': 0}, {\'item_id\': \'3170\', \'item_name\': "King\'s Bounty: Armored Princess", \'playtime_forever\': 1021, \'playtime_2weeks\': 0}, {\'item_id\': \'25900\', \'item_name\': "King\'s Bounty: The Legend", \'playtime_forever\': 1389, \'playtime_2weeks\': 0}, {\'item_id\': \'31410\', \'item_name\': \'Zombie Driver\', \'playtime_forever\': 95, \'playtime_2weeks\': 0}, {\'item_id\': \'24980\', \'item_name\': \'Mass Effect 2\', \'playtime_forever\': 5001, \'playtime_2weeks\': 0}, {\'item_id\': \'8850\', \'item_name\': \'BioShock 2\', \'playtime_forever\': 1504, \'playtime_2weeks\': 0}, {\'item_id\': \'409720\', \'item_name\': \'BioShock 2 Remastered\', \'playtime_forever\': 0, \'playtime_2weeks\': 0}, {\'item_id\': \'46000\', \'item_name\': \'Bob Came in Pieces\', \'playtime_forever\': 11, \'playtime_2weeks\': 0}, {\'item_id\': \'33230\', \'item_name\': "Assassin\'s Creed II", \'playtime_forever\': 1636, \'playtime_2weeks\': 0}, {\'item_id\': \'20820\', \'item_name\': \'Shatter\', \'playtime_forever\': 46, \'playtime_2weeks\': 0}, {\'item_id\': \'47700\', \'item_name\': \'Command and Conquer 4: Tiberian Twilight\', \'playtime_forever\': 85, \'playtime_2weeks\': 0}, {\'item_id\': \'24960\', \'item_name\': \'Battlefield: Bad Company 2\', \'playtime_forever\': 5716, \'playtime_2weeks\': 0}, {\'item_id\': \'43110\', \'item_name\': \'Metro 2033\', \'playtime_forever\': 834, \'playtime_2weeks\': 0}, {\'item_id\': \'8190\', \'item_name\': \'Just Cause 2\', \'playtime_forever\': 3083, \'playtime_2weeks\': 0}, {\'item_id\': \'49600\', \'item_name\': \'Beat Hazard\', \'playtime_forever\': 127, \'playtime_2weeks\': 0}, {\'item_id\': \'31220\', \'item_name\': \'Sam & Max 301: The Penal Zone\', \'playtime_forever\': 71, \'playtime_2weeks\': 0}, {\'item_id\': \'31230\', \'item_name\': \'Sam & Max 302: The Tomb of Sammun-Mak\', \'playtime_forever\': 0, \'playtime_2weeks\': 0}, {\'item_id\': \'31240\', \'item_name\': "Sam & Max 303: They Stole Max\'s Brain!", \'playtime_forever\': 0, \'playtime_2weeks\': 0}, {\'item_id\': \'31250\', \'item_name\': \'Sam & Max 304: Beyond the Alley of the Dolls\', \'playtime_forever\': 0, \'playtime_2weeks\': 0}, {\'item_id\': \'31260\', \'item_name\': \'Sam & Max 305: The City that Dares not Sleep\', \'playtime_forever\': 0, \'playtime_2weeks\': 0}, {\'item_id\': \'21090\', \'item_name\': \'F.E.A.R.\', \'playtime_forever\': 585, \'playtime_2weeks\': 0}, {\'item_id\': \'21110\', \'item_name\': \'F.E.A.R.: Extraction Point\', \'playtime_forever\': 146, \'playtime_2weeks\': 0}, {\'item_id\': \'21120\', \'item_name\': \'F.E.A.R.: Perseus Mandate\', \'playtime_forever\': 0, \'playtime_2weeks\': 0}, {\'item_id\': \'49400\', \'item_name\': \'Magic: The Gathering - Duels of the Planeswalkers\', \'playtime_forever\': 32, \'playtime_2weeks\': 0}, {\'item_id\': \'33910\', \'item_name\': \'Arma 2\', \'playtime_forever\': 3, \'playtime_2weeks\': 0}, {\'item_id\': \'33930\',

\'item_name\': \'Arma 2: Operation Arrowhead\', \'playtime_forever\': 3,
\'playtime_2weeks\': 0}, {\'item_id\': \'219540\', \'item_name\': \'Arma 2:
Operation Arrowhead Beta (Obsolete)\', \'playtime_forever\': 0,
\'playtime_2weeks\': 0}, {\'item_id\': \'17410\', \'item_name\': "Mirror\'s
Edge", \'playtime_forever\': 0, \'playtime_2weeks\': 0}, {\'item_id\':
\'205790\', \'item_name\': \'Dota 2 Test\', \'playtime_forever\': 0,
\'playtime_2weeks\': 0}, {\'item_id\': \'50130\', \'item_name\': \'Mafia II\',
\'playtime_forever\': 707, \'playtime_2weeks\': 0}, {\'item_id\': \'62100\',
\'item_name\': \'Chime\', \'playtime_forever\': 50, \'playtime_2weeks\': 0},
{\'item_id\': \'31280\', \'item_name\': \'Poker Night at the Inventory\',
\'playtime_forever\': 206, \'playtime_2weeks\': 0}, {\'item_id\': \'57300\',
\'item_name\': \'Amnesia: The Dark Descent\', \'playtime_forever\': 12,
\'playtime_2weeks\': 0}, {\'item_id\': \'21970\', \'item_name\': \'R.U.S.E\',
\'playtime_forever\': 64, \'playtime_2weeks\': 0}, {\'item_id\': \'8930\',
\'item_name\': "Sid Meier\'s Civilization V", \'playtime_forever\': 10345,
\'playtime_2weeks\': 0}, {\'item_id\': \'41000\', \'item_name\': \'Serious Sam
HD: The First Encounter\', \'playtime_forever\': 369, \'playtime_2weeks\': 0},
{\'item_id\': \'41050\', \'item_name\': \'Serious Sam Classic: The First
Encounter\', \'playtime_forever\': 76, \'playtime_2weeks\': 0}, {\'item_id\':
\'41060\', \'item_name\': \'Serious Sam Classic: The Second Encounter\',
\'playtime_forever\': 0, \'playtime_2weeks\': 0}, {\'item_id\': \'227780\',
\'item_name\': \'Serious Sam Classics: Revolution\', \'playtime_forever\': 0,
\'playtime_2weeks\': 0}, {\'item_id\': \'18040\', \'item_name\': \'DeathSpank\',
\'playtime_forever\': 594, \'playtime_2weeks\': 0}, {\'item_id\': \'22380\',
\'item_name\': \'Fallout: New Vegas\', \'playtime_forever\': 43,
\'playtime_2weeks\': 0}, {\'item_id\': \'42700\', \'item_name\': \'Call of Duty:
Black Ops\', \'playtime_forever\': 966, \'playtime_2weeks\': 0}, {\'item_id\':
\'42710\', \'item_name\': \'Call of Duty: Black Ops - Multiplayer\',
\'playtime_forever\': 2452, \'playtime_2weeks\': 0}, {\'item_id\': \'62000\',
\'item_name\': \'Flight Control HD\', \'playtime_forever\': 25,
\'playtime_2weeks\': 0}, {\'item_id\': \'40800\', \'item_name\': \'Super Meat
Boy\', \'playtime_forever\': 119, \'playtime_2weeks\': 0}, {\'item_id\':
\'9500\', \'item_name\': \'Gish\', \'playtime_forever\': 6, \'playtime_2weeks\':
0}, {\'item_id\': \'18700\', \'item_name\': \'And Yet It Moves\',
\'playtime_forever\': 0, \'playtime_2weeks\': 0}, {\'item_id\': \'31270\',
\'item_name\': \'Puzzle Agent\', \'playtime_forever\': 0, \'playtime_2weeks\':
0}, {\'item_id\': \'58200\', \'item_name\': \'Jolly Rover\',
\'playtime_forever\': 0, \'playtime_2weeks\': 0}, {\'item_id\': \'70400\',
\'item_name\': "Recettear: An Item Shop\'s Tale", \'playtime_forever\': 537,
\'playtime_2weeks\': 0}, {\'item_id\': \'41300\', \'item_name\': \'Altitude\',
\'playtime_forever\': 50, \'playtime_2weeks\': 0}, {\'item_id\': \'41800\',
\'item_name\': \'Gratuitous Space Battles\', \'playtime_forever\': 149,
\'playtime_2weeks\': 0}, {\'item_id\': \'42500\', \'item_name\': \'DogFighter\',
\'playtime_forever\': 0, \'playtime_2weeks\': 0}, {\'item_id\': \'49900\',
\'item_name\': \'Plain Sight\', \'playtime_forever\': 23, \'playtime_2weeks\':
0}, {\'item_id\': \'55000\', \'item_name\': \'Flotilla\', \'playtime_forever\':
0, \'playtime_2weeks\': 0}, {\'item_id\': \'34330\', \'item_name\': \'Total War:

SHOGUN 2\', \'playtime_forever\': 1538, \'playtime_2weeks\': 0}, {\'item_id\':
\'63200\', \'item_name\': \'Monday Night Combat\', \'playtime_forever\': 573,
\'playtime_2weeks\': 0}, {\'item_id\': \'99810\', \'item_name\':
\'Bulletstorm\', \'playtime_forever\': 2, \'playtime_2weeks\': 0}, {\'item_id\':
\'620\', \'item_name\': \'Portal 2\', \'playtime_forever\': 887,
\'playtime_2weeks\': 0}, {\'item_id\': \'42910\', \'item_name\': \'Magicka\',
\'playtime_forever\': 588, \'playtime_2weeks\': 0}, {\'item_id\': \'15500\',
\'item_name\': \'The Wonderful End of the World\', \'playtime_forever\': 0,
\'playtime_2weeks\': 0}, {\'item_id\': \'15540\', \'item_name\': \'1… 2…
3… KICK IT! (Drop That Beat Like an Ugly Baby)\', \'playtime_forever\': 4,
\'playtime_2weeks\': 0}, {\'item_id\': \'18500\', \'item_name\': \'Defense Grid:
The Awakening\', \'playtime_forever\': 126, \'playtime_2weeks\': 0},
{\'item_id\': \'26500\', \'item_name\': \'Cogs\', \'playtime_forever\': 0,
\'playtime_2weeks\': 0}, {\'item_id\': \'35460\', \'item_name\': \'The Ball\',
\'playtime_forever\': 0, \'playtime_2weeks\': 0}, {\'item_id\': \'38700\',
\'item_name\': \'Toki Tori\', \'playtime_forever\': 0, \'playtime_2weeks\': 0},
{\'item_id\': \'38720\', \'item_name\': \'RUSH\', \'playtime_forever\': 0,
\'playtime_2weeks\': 0}, {\'item_id\': \'63700\', \'item_name\': \'BIT.TRIP
BEAT\', \'playtime_forever\': 0, \'playtime_2weeks\': 0}, {\'item_id\': \'280\',
\'item_name\': \'Half-Life: Source\', \'playtime_forever\': 0,
\'playtime_2weeks\': 0}, {\'item_id\': \'20920\', \'item_name\': \'The Witcher
2: Assassins of Kings Enhanced Edition\', \'playtime_forever\': 1343,
\'playtime_2weeks\': 0}, {\'item_id\': \'105600\', \'item_name\': \'Terraria\',
\'playtime_forever\': 772, \'playtime_2weeks\': 0}, {\'item_id\': \'98200\',
\'item_name\': \'Frozen Synapse\', \'playtime_forever\': 9, \'playtime_2weeks\':
0}, {\'item_id\': \'4500\', \'item_name\': \'S.T.A.L.K.E.R.: Shadow of
Chernobyl\', \'playtime_forever\': 0, \'playtime_2weeks\': 0}, {\'item_id\':
\'4540\', \'item_name\': \'Titan Quest\', \'playtime_forever\': 3,
\'playtime_2weeks\': 0}, {\'item_id\': \'4550\', \'item_name\': \'Titan Quest:
Immortal Throne\', \'playtime_forever\': 76, \'playtime_2weeks\': 0},
{\'item_id\': \'9480\', \'item_name\': \'Saints Row 2\', \'playtime_forever\':
0, \'playtime_2weeks\': 0}, {\'item_id\': \'15620\', \'item_name\': \'Warhammer®
40,000 : Dawn of War® II\', \'playtime_forever\': 2085, \'playtime_2weeks\': 0},
{\'item_id\': \'20500\', \'item_name\': \'Red Faction: Guerrilla Steam
Edition\', \'playtime_forever\': 0, \'playtime_2weeks\': 0}, {\'item_id\':
\'20570\', \'item_name\': \'Warhammer® 40,000 : Dawn of War® II - Chaos
Rising \', \'playtime_forever\': 69, \'playtime_2weeks\': 0}, {\'item_id\':
\'50620\', \'item_name\': \'Darksiders\', \'playtime_forever\': 101,
\'playtime_2weeks\': 0}, {\'item_id\': \'55100\', \'item_name\': \'Homefront\',
\'playtime_forever\': 51, \'playtime_2weeks\': 0}, {\'item_id\': \'55140\',
\'item_name\': \'MX vs. ATV Reflex\', \'playtime_forever\': 0,
\'playtime_2weeks\': 0}, {\'item_id\': \'56400\', \'item_name\': \'Warhammer®
40,000 : Dawn of War® II - Retribution \', \'playtime_forever\': 678,
\'playtime_2weeks\': 0}, {\'item_id\': \'475150\', \'item_name\': \'Titan Quest
Anniversary Edition\', \'playtime_forever\': 0, \'playtime_2weeks\': 0},
{\'item_id\': \'96100\', \'item_name\': \'Defy Gravity\', \'playtime_forever\':
9, \'playtime_2weeks\': 0}, {\'item_id\': \'35450\', \'item_name\': \'Rising

Storm/Red Orchestra 2 Multiplayer\', \'playtime_forever\': 14194,
\'playtime_2weeks\': 0}, {\'item_id\': \'236830\', \'item_name\': \'Red
Orchestra 2: Heroes of Stalingrad - Single Player\', \'playtime_forever\': 0,
\'playtime_2weeks\': 0}, {\'item_id\': \'72850\', \'item_name\': \'The Elder
Scrolls V: Skyrim\', \'playtime_forever\': 2512, \'playtime_2weeks\': 0},
{\'item_id\': \'107100\', \'item_name\': \'Bastion\', \'playtime_forever\': 640,
\'playtime_2weeks\': 0}, {\'item_id\': \'57690\', \'item_name\': \'Tropico 4\',
\'playtime_forever\': 1815, \'playtime_2weeks\': 0}, {\'item_id\': \'91310\',
\'item_name\': \'Dead Island\', \'playtime_forever\': 2078, \'playtime_2weeks\':
0}, {\'item_id\': \'113200\', \'item_name\': \'The Binding of Isaac\',
\'playtime_forever\': 1494, \'playtime_2weeks\': 0}, {\'item_id\': \'65800\',
\'item_name\': \'Dungeon Defenders\', \'playtime_forever\': 130,
\'playtime_2weeks\': 0}, {\'item_id\': \'9200\', \'item_name\': \'RAGE\',
\'playtime_forever\': 653, \'playtime_2weeks\': 0}, {\'item_id\': \'41070\',
\'item_name\': \'Serious Sam 3: BFE\', \'playtime_forever\': 716,
\'playtime_2weeks\': 0}, {\'item_id\': \'48950\', \'item_name\': \'Greed Corp\',
\'playtime_forever\': 72, \'playtime_2weeks\': 0}, {\'item_id\': \'92800\',
\'item_name\': \'SpaceChem\', \'playtime_forever\': 262, \'playtime_2weeks\':
0}, {\'item_id\': \'104700\', \'item_name\': \'Super Monday Night Combat\',
\'playtime_forever\': 44, \'playtime_2weeks\': 0}, {\'item_id\': \'203770\',
\'item_name\': \'Crusader Kings II\', \'playtime_forever\': 18,
\'playtime_2weeks\': 0}, {\'item_id\': \'203810\', \'item_name\': \'Dear
Esther\', \'playtime_forever\': 86, \'playtime_2weeks\': 0}, {\'item_id\':
\'110800\', \'item_name\': \'L.A. Noire\', \'playtime_forever\': 1204,
\'playtime_2weeks\': 0}, {\'item_id\': \'207610\', \'item_name\': \'The Walking
Dead\', \'playtime_forever\': 798, \'playtime_2weeks\': 0}, {\'item_id\':
\'98300\', \'item_name\': \'Toy Soldiers\', \'playtime_forever\': 110,
\'playtime_2weeks\': 0}, {\'item_id\': \'50300\', \'item_name\': \'Spec Ops: The
Line\', \'playtime_forever\': 296, \'playtime_2weeks\': 0}, {\'item_id\':
\'204300\', \'item_name\': \'Awesomenauts\', \'playtime_forever\': 0,
\'playtime_2weeks\': 0}, {\'item_id\': \'1700\', \'item_name\': \'Arx Fatalis\',
\'playtime_forever\': 0, \'playtime_2weeks\': 0}, {\'item_id\': \'205100\',
\'item_name\': \'Dishonored\', \'playtime_forever\': 1429, \'playtime_2weeks\':
0}, {\'item_id\': \'211120\', \'item_name\': \'The Political Machine\',
\'playtime_forever\': 1, \'playtime_2weeks\': 0}, {\'item_id\': \'730\',
\'item_name\': \'Counter-Strike: Global Offensive\', \'playtime_forever\':
23532, \'playtime_2weeks\': 0}, {\'item_id\': \'49520\', \'item_name\':
\'Borderlands 2\', \'playtime_forever\': 1303, \'playtime_2weeks\': 0},
{\'item_id\': \'200260\', \'item_name\': \'Batman: Arkham City GOTY\',
\'playtime_forever\': 100, \'playtime_2weeks\': 0}, {\'item_id\': \'212680\',
\'item_name\': \'FTL: Faster Than Light\', \'playtime_forever\': 500,
\'playtime_2weeks\': 0}, {\'item_id\': \'200710\', \'item_name\': \'Torchlight
II\', \'playtime_forever\': 459, \'playtime_2weeks\': 0}, {\'item_id\':
\'219150\', \'item_name\': \'Hotline Miami\', \'playtime_forever\': 0,
\'playtime_2weeks\': 0}, {\'item_id\': \'215510\', \'item_name\': \'Rocketbirds:
Hardboiled Chicken\', \'playtime_forever\': 0, \'playtime_2weeks\': 0},
{\'item_id\': \'8870\', \'item_name\': \'BioShock Infinite\',

\'playtime_forever\': 2084, \'playtime_2weeks\': 0}, {\'item_id\': \'200510\',
\'item_name\': \'XCOM: Enemy Unknown\', \'playtime_forever\': 3478,
\'playtime_2weeks\': 0}, {\'item_id\': \'220240\', \'item_name\': \'Far Cry®
3\', \'playtime_forever\': 182, \'playtime_2weeks\': 0}, {\'item_id\':
\'218680\', \'item_name\': \'Scribblenauts Unlimited\', \'playtime_forever\': 0,
\'playtime_2weeks\': 0}, {\'item_id\': \'228280\', \'item_name\': "Baldur\'s
Gate: Enhanced Edition", \'playtime_forever\': 905, \'playtime_2weeks\': 0},
{\'item_id\': \'212070\', \'item_name\': \'Star Conflict\',
\'playtime_forever\': 0, \'playtime_2weeks\': 0}, {\'item_id\': \'218230\',
\'item_name\': \'PlanetSide 2\', \'playtime_forever\': 51, \'playtime_2weeks\':
0}, {\'item_id\': \'236390\', \'item_name\': \'War Thunder\',
\'playtime_forever\': 50, \'playtime_2weeks\': 0}, {\'item_id\': \'3483\',
\'item_name\': \'Peggle Extreme\', \'playtime_forever\': 0, \'playtime_2weeks\':
0}, {\'item_id\': \'203160\', \'item_name\': \'Tomb Raider\',
\'playtime_forever\': 0, \'playtime_2weeks\': 0}, {\'item_id\': \'227080\',
\'item_name\': \'Papo & Yo\', \'playtime_forever\': 81, \'playtime_2weeks\': 0},
{\'item_id\': \'233450\', \'item_name\': \'Prison Architect\',
\'playtime_forever\': 24, \'playtime_2weeks\': 0}, {\'item_id\': \'43160\',
\'item_name\': \'Metro: Last Light\', \'playtime_forever\': 712,
\'playtime_2weeks\': 0}, {\'item_id\': \'219740\', \'item_name\': "Don\'t
Starve", \'playtime_forever\': 139, \'playtime_2weeks\': 0}, {\'item_id\':
\'322330\', \'item_name\': "Don\'t Starve Together", \'playtime_forever\': 0,
\'playtime_2weeks\': 0}, {\'item_id\': \'222730\', \'item_name\': \'Reus\',
\'playtime_forever\': 319, \'playtime_2weeks\': 0}, {\'item_id\': \'241600\',
\'item_name\': \'Rogue Legacy\', \'playtime_forever\': 73, \'playtime_2weeks\':
0}, {\'item_id\': \'238930\', \'item_name\': \'7 Grand Steps, Step 1: What
Ancients Begat\', \'playtime_forever\': 159, \'playtime_2weeks\': 0},
{\'item_id\': \'217140\', \'item_name\': \'Rise of the Triad\',
\'playtime_forever\': 199, \'playtime_2weeks\': 0}, {\'item_id\': \'39140\',
\'item_name\': \'FINAL FANTASY VII\', \'playtime_forever\': 2896,
\'playtime_2weeks\': 0}, {\'item_id\': \'225080\', \'item_name\': \'Brothers - A
Tale of Two Sons\', \'playtime_forever\': 0, \'playtime_2weeks\': 0},
{\'item_id\': \'239030\', \'item_name\': \'Papers, Please\',
\'playtime_forever\': 194, \'playtime_2weeks\': 0}, {\'item_id\': \'248820\',
\'item_name\': \'Risk of Rain\', \'playtime_forever\': 272, \'playtime_2weeks\':
0}, {\'item_id\': \'239350\', \'item_name\': \'Spelunky\', \'playtime_forever\':
458, \'playtime_2weeks\': 0}, {\'item_id\': \'218620\', \'item_name\': \'PAYDAY
2\', \'playtime_forever\': 574, \'playtime_2weeks\': 0}, {\'item_id\':
\'206420\', \'item_name\': \'Saints Row IV\', \'playtime_forever\': 0,
\'playtime_2weeks\': 0}, {\'item_id\': \'236850\', \'item_name\': \'Europa
Universalis IV\', \'playtime_forever\': 89, \'playtime_2weeks\': 0},
{\'item_id\': \'47790\', \'item_name\': \'Medal of Honor(TM) Single Player\',
\'playtime_forever\': 230, \'playtime_2weeks\': 0}, {\'item_id\': \'47830\',
\'item_name\': \'Medal of Honor(TM) Multiplayer\', \'playtime_forever\': 0,
\'playtime_2weeks\': 0}, {\'item_id\': \'250320\', \'item_name\': \'The Wolf
Among Us\', \'playtime_forever\': 500, \'playtime_2weeks\': 0}, {\'item_id\':
\'250900\', \'item_name\': \'The Binding of Isaac: Rebirth\',

\'playtime_forever\': 329, \'playtime_2weeks\': 0}, {\'item_id\': \'204450\',
\'item_name\': \'Call of Juarez Gunslinger\', \'playtime_forever\': 241,
\'playtime_2weeks\': 0}, {\'item_id\': \'233270\', \'item_name\': \'Far Cry® 3
Blood Dragon\', \'playtime_forever\': 0, \'playtime_2weeks\': 0}, {\'item_id\':
\'251990\', \'item_name\': \'Long Live The Queen\', \'playtime_forever\': 135,
\'playtime_2weeks\': 0}, {\'item_id\': \'253980\', \'item_name\': \'Enclave\',
\'playtime_forever\': 0, \'playtime_2weeks\': 0}, {\'item_id\': \'236450\',
\'item_name\': \'PAC-MAN Championship Edition DX+\', \'playtime_forever\': 154,
\'playtime_2weeks\': 0}, {\'item_id\': \'242050\', \'item_name\': "Assassin\'s
Creed IV Black Flag", \'playtime_forever\': 1377, \'playtime_2weeks\': 0},
{\'item_id\': \'257350\', \'item_name\': "Baldur\'s Gate II: Enhanced Edition",
\'playtime_forever\': 0, \'playtime_2weeks\': 0}, {\'item_id\': \'213670\',
\'item_name\': \'South Park : The Stick of Truth \', \'playtime_forever\': 121,
\'playtime_2weeks\': 0}, {\'item_id\': \'238010\', \'item_name\': "Deus Ex:
Human Revolution - Director\'s Cut", \'playtime_forever\': 2176,
\'playtime_2weeks\': 0}, {\'item_id\': \'245470\', \'item_name\': \'Democracy
3\', \'playtime_forever\': 246, \'playtime_2weeks\': 0}, {\'item_id\':
\'221910\', \'item_name\': \'The Stanley Parable\', \'playtime_forever\': 157,
\'playtime_2weeks\': 0}, {\'item_id\': \'260230\', \'item_name\': \'Valiant
Hearts: The Great War / Soldats Inconnus : Mémoires de la Grande Guerre \',
\'playtime_forever\': 467, \'playtime_2weeks\': 0}, {\'item_id\': \'238430\',
\'item_name\': \'Contagion\', \'playtime_forever\': 77, \'playtime_2weeks\': 0},
{\'item_id\': \'261030\', \'item_name\': \'The Walking Dead: Season Two\',
\'playtime_forever\': 567, \'playtime_2weeks\': 0}, {\'item_id\': \'261570\',
\'item_name\': \'Ori and the Blind Forest\', \'playtime_forever\': 39,
\'playtime_2weeks\': 0}, {\'item_id\': \'263980\', \'item_name\': \'Out There
Somewhere\', \'playtime_forever\': 0, \'playtime_2weeks\': 0}, {\'item_id\':
\'39150\', \'item_name\': \'FINAL FANTASY VIII\', \'playtime_forever\': 15,
\'playtime_2weeks\': 0}, {\'item_id\': \'230230\', \'item_name\': \'Divinity:
Original Sin (Classic)\', \'playtime_forever\': 522, \'playtime_2weeks\': 0},
{\'item_id\': \'373420\', \'item_name\': \'Divinity: Original Sin Enhanced
Edition\', \'playtime_forever\': 0, \'playtime_2weeks\': 0}, {\'item_id\':
\'232790\', \'item_name\': \'Broken Age\', \'playtime_forever\': 216,
\'playtime_2weeks\': 0}, {\'item_id\': \'237990\', \'item_name\': \'The Banner
Saga\', \'playtime_forever\': 19, \'playtime_2weeks\': 0}, {\'item_id\':
\'273620\', \'item_name\': \'Freespace 2\', \'playtime_forever\': 848,
\'playtime_2weeks\': 0}, {\'item_id\': \'222880\', \'item_name\':
\'Insurgency\', \'playtime_forever\': 3031, \'playtime_2weeks\': 0},
{\'item_id\': \'225840\', \'item_name\': \'Sven Co-op\', \'playtime_forever\':
26, \'playtime_2weeks\': 0}, {\'item_id\': \'222900\', \'item_name\': \'Dead
Island: Epidemic\', \'playtime_forever\': 0, \'playtime_2weeks\': 0},
{\'item_id\': \'280220\', \'item_name\': \'Creeper World 3: Arc Eternal\',
\'playtime_forever\': 941, \'playtime_2weeks\': 0}, {\'item_id\': \'236430\',
\'item_name\': \'DARK SOULS II\', \'playtime_forever\': 592,
\'playtime_2weeks\': 0}, {\'item_id\': \'201810\', \'item_name\': \'Wolfenstein:
The New Order\', \'playtime_forever\': 599, \'playtime_2weeks\': 0},
{\'item_id\': \'294860\', \'item_name\': \'Valkyria Chronicles \',

\'playtime_forever\': 1735, \'playtime_2weeks\': 0}, {\'item_id\': \'295690\',
\'item_name\': \'PARTICLE MACE\', \'playtime_forever\': 0, \'playtime_2weeks\':
0}, {\'item_id\': \'238460\', \'item_name\': \'BattleBlock Theater\',
\'playtime_forever\': 407, \'playtime_2weeks\': 0}, {\'item_id\': \'237930\',
\'item_name\': \'Transistor\', \'playtime_forever\': 154, \'playtime_2weeks\':
0}, {\'item_id\': \'301480\', \'item_name\': \'GEARCRACK Arena\',
\'playtime_forever\': 0, \'playtime_2weeks\': 0}, {\'item_id\': \'303390\',
\'item_name\': \'Dead Bits\', \'playtime_forever\': 0, \'playtime_2weeks\': 0},
{\'item_id\': \'306660\', \'item_name\': \'Ultimate General: Gettysburg\',
\'playtime_forever\': 1245, \'playtime_2weeks\': 0}, {\'item_id\': \'316790\',
\'item_name\': \'Grim Fandango Remastered\', \'playtime_forever\': 42,
\'playtime_2weeks\': 0}, {\'item_id\': \'318430\', \'item_name\': \'Squishy the
Suicidal Pig\', \'playtime_forever\': 0, \'playtime_2weeks\': 0}, {\'item_id\':
\'241930\', \'item_name\': \'Middle-earth: Shadow of Mordor\',
\'playtime_forever\': 889, \'playtime_2weeks\': 0}, {\'item_id\': \'328080\',
\'item_name\': \'Retro-Pixel Castles\', \'playtime_forever\': 5,
\'playtime_2weeks\': 0}, {\'item_id\': \'282070\', \'item_name\': \'This War of
Mine\', \'playtime_forever\': 3, \'playtime_2weeks\': 0}, {\'item_id\':
\'333930\', \'item_name\': \'Dirty Bomb\', \'playtime_forever\': 2782,
\'playtime_2weeks\': 0}, {\'item_id\': \'334120\', \'item_name\': \'Armikrog\',
\'playtime_forever\': 109, \'playtime_2weeks\': 0}, {\'item_id\': \'257510\',
\'item_name\': \'The Talos Principle\', \'playtime_forever\': 0,
\'playtime_2weeks\': 0}, {\'item_id\': \'319630\', \'item_name\': \'Life Is
Strange \', \'playtime_forever\': 37, \'playtime_2weeks\': 0}, {\'item_id\':
\'346010\', \'item_name\': \'Besiege\', \'playtime_forever\': 464,
\'playtime_2weeks\': 0}, {\'item_id\': \'255710\', \'item_name\': \'Cities:
Skylines\', \'playtime_forever\': 246, \'playtime_2weeks\': 0}, {\'item_id\':
\'232090\', \'item_name\': \'Killing Floor 2\', \'playtime_forever\': 6494,
\'playtime_2weeks\': 0}, {\'item_id\': \'232150\', \'item_name\': \'Killing
Floor 2 - SDK\', \'playtime_forever\': 0, \'playtime_2weeks\': 0}, {\'item_id\':
\'377160\', \'item_name\': \'Fallout 4\', \'playtime_forever\': 2001,
\'playtime_2weeks\': 0}, {\'item_id\': \'340170\', \'item_name\': \'FINAL
FANTASY TYPE-0 HD\', \'playtime_forever\': 170, \'playtime_2weeks\': 0},
{\'item_id\': \'391540\', \'item_name\': \'Undertale\', \'playtime_forever\':
395, \'playtime_2weeks\': 0}, {\'item_id\': \'396660\', \'item_name\': \'Tahira:
Echoes of the Astral Empire\', \'playtime_forever\': 0, \'playtime_2weeks\': 0},
{\'item_id\': \'337000\', \'item_name\': \'Deus Ex: Mankind Divided \',
\'playtime_forever\': 1159, \'playtime_2weeks\': 0}, {\'item_id\': \'368230\',
\'item_name\': \'Kingdom: Classic\', \'playtime_forever\': 83,
\'playtime_2weeks\': 0}, {\'item_id\': \'496300\', \'item_name\': \'Kingdom: New
Lands\', \'playtime_forever\': 0, \'playtime_2weeks\': 0}, {\'item_id\':
\'231430\', \'item_name\': \'Company of Heroes 2\', \'playtime_forever\': 0,
\'playtime_2weeks\': 0}, {\'item_id\': \'289130\', \'item_name\': \'Endless
Legend\', \'playtime_forever\': 593, \'playtime_2weeks\': 0}, {\'item_id\':
\'268500\', \'item_name\': \'XCOM 2\', \'playtime_forever\': 1495,
\'playtime_2weeks\': 0}, {\'item_id\': \'461640\', \'item_name\': \'Sins Of The
Demon RPG\', \'playtime_forever\': 0, \'playtime_2weeks\': 0}, {\'item_id\':

```
\'379720\', \'item_name\': \'DOOM\', \'playtime_forever\': 793,
\'playtime_2weeks\': 0}, {\'item_id\': \'273350\', \'item_name\': \'Evolve Stage
2\', \'playtime_forever\': 58, \'playtime_2weeks\': 0}]}\n'
```

[ ]:
```
len(lines)
```

[ ]:
```
88310
```

Total 10000 lines in data , Every Line display Player Data

[ ]:
```
data_linter = ast.literal_eval(lines[0])
```

[ ]:
```
data_linter
```

[ ]:
```
{'user_id': '76561197970982479',
 'items_count': 277,
 'steam_id': '76561197970982479',
 'user_url': 'http://steamcommunity.com/profiles/76561197970982479',
 'items': [{'item_id': '10',
   'item_name': 'Counter-Strike',
   'playtime_forever': 6,
   'playtime_2weeks': 0},
  {'item_id': '20',
   'item_name': 'Team Fortress Classic',
   'playtime_forever': 0,
   'playtime_2weeks': 0},
  {'item_id': '30',
   'item_name': 'Day of Defeat',
   'playtime_forever': 7,
   'playtime_2weeks': 0},
  {'item_id': '40',
   'item_name': 'Deathmatch Classic',
   'playtime_forever': 0,
   'playtime_2weeks': 0},
  {'item_id': '50',
   'item_name': 'Half-Life: Opposing Force',
   'playtime_forever': 0,
   'playtime_2weeks': 0},
  {'item_id': '60',
   'item_name': 'Ricochet',
   'playtime_forever': 0,
   'playtime_2weeks': 0},
  {'item_id': '70',
   'item_name': 'Half-Life',
   'playtime_forever': 0,
   'playtime_2weeks': 0},
  {'item_id': '130',
   'item_name': 'Half-Life: Blue Shift',
```

```
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '300',
 'item_name': 'Day of Defeat: Source',
 'playtime_forever': 4733,
 'playtime_2weeks': 0},
{'item_id': '240',
 'item_name': 'Counter-Strike: Source',
 'playtime_forever': 1853,
 'playtime_2weeks': 0},
{'item_id': '3830',
 'item_name': 'Psychonauts',
 'playtime_forever': 333,
 'playtime_2weeks': 0},
{'item_id': '2630',
 'item_name': 'Call of Duty 2',
 'playtime_forever': 75,
 'playtime_2weeks': 0},
{'item_id': '3900',
 'item_name': "Sid Meier's Civilization IV",
 'playtime_forever': 338,
 'playtime_2weeks': 0},
{'item_id': '34440',
 'item_name': "Sid Meier's Civilization IV",
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '3920',
 'item_name': "Sid Meier's Pirates!",
 'playtime_forever': 2,
 'playtime_2weeks': 0},
{'item_id': '6400',
 'item_name': 'Joint Task Force',
 'playtime_forever': 286,
 'playtime_2weeks': 0},
{'item_id': '6910',
 'item_name': 'Deus Ex: Game of the Year Edition',
 'playtime_forever': 2685,
 'playtime_2weeks': 0},
{'item_id': '7670',
 'item_name': 'BioShock',
 'playtime_forever': 633,
 'playtime_2weeks': 0},
{'item_id': '409710',
 'item_name': 'BioShock Remastered',
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '220',
```

```
 'item_name': 'Half-Life 2',
 'playtime_forever': 696,
 'playtime_2weeks': 0},
{'item_id': '320',
 'item_name': 'Half-Life 2: Deathmatch',
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '340',
 'item_name': 'Half-Life 2: Lost Coast',
 'playtime_forever': 37,
 'playtime_2weeks': 0},
{'item_id': '360',
 'item_name': 'Half-Life Deathmatch: Source',
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '380',
 'item_name': 'Half-Life 2: Episode One',
 'playtime_forever': 168,
 'playtime_2weeks': 0},
{'item_id': '400',
 'item_name': 'Portal',
 'playtime_forever': 173,
 'playtime_2weeks': 0},
{'item_id': '420',
 'item_name': 'Half-Life 2: Episode Two',
 'playtime_forever': 323,
 'playtime_2weeks': 0},
{'item_id': '9340',
 'item_name': 'Company of Heroes: Opposing Fronts',
 'playtime_forever': 692,
 'playtime_2weeks': 0},
{'item_id': '228200',
 'item_name': 'Company of Heroes (New Steam Version)',
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '11450',
 'item_name': 'Overlord',
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '7940',
 'item_name': 'Call of Duty 4: Modern Warfare',
 'playtime_forever': 1185,
 'playtime_2weeks': 0},
{'item_id': '4700',
 'item_name': 'Medieval II: Total War',
 'playtime_forever': 477,
 'playtime_2weeks': 0},
```

```
{'item_id': '12900',
 'item_name': 'Audiosurf',
 'playtime_forever': 115,
 'playtime_2weeks': 0},
{'item_id': '13250',
 'item_name': 'Unreal Gold',
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '16100',
 'item_name': 'Virtual Villagers: A New Home',
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '15700',
 'item_name': "Oddworld: Abe's Oddysee",
 'playtime_forever': 28,
 'playtime_2weeks': 0},
{'item_id': '15710',
 'item_name': "Oddworld: Abe's Exoddus",
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '17330',
 'item_name': 'Crysis Warhead',
 'playtime_forever': 31,
 'playtime_2weeks': 0},
{'item_id': '17340',
 'item_name': 'Crysis Wars',
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '22000',
 'item_name': 'World of Goo',
 'playtime_forever': 195,
 'playtime_2weeks': 0},
{'item_id': '500',
 'item_name': 'Left 4 Dead',
 'playtime_forever': 513,
 'playtime_2weeks': 0},
{'item_id': '4560',
 'item_name': 'Company of Heroes',
 'playtime_forever': 1061,
 'playtime_2weeks': 0},
{'item_id': '17460',
 'item_name': 'Mass Effect',
 'playtime_forever': 1613,
 'playtime_2weeks': 0},
{'item_id': '10500',
 'item_name': 'Empire: Total War',
 'playtime_forever': 186,
```

```
 'playtime_2weeks': 0},
{'item_id': '24740',
 'item_name': 'Burnout Paradise: The Ultimate Box',
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '22200',
 'item_name': 'Zeno Clash',
 'playtime_forever': 271,
 'playtime_2weeks': 0},
{'item_id': '26800',
 'item_name': 'Braid',
 'playtime_forever': 445,
 'playtime_2weeks': 0},
{'item_id': '1250',
 'item_name': 'Killing Floor',
 'playtime_forever': 10006,
 'playtime_2weeks': 0},
{'item_id': '35420',
 'item_name': 'Killing Floor Mod: Defence Alliance 2',
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '3590',
 'item_name': 'Plants vs. Zombies: Game of the Year',
 'playtime_forever': 4413,
 'playtime_2weeks': 0},
{'item_id': '8880',
 'item_name': 'Freedom Force',
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '8890',
 'item_name': 'Freedom Force vs. the 3rd Reich',
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '23120',
 'item_name': 'Droplitz',
 'playtime_forever': 53,
 'playtime_2weeks': 0},
{'item_id': '35700',
 'item_name': 'Trine',
 'playtime_forever': 199,
 'playtime_2weeks': 0},
{'item_id': '10140',
 'item_name': '3D Ultra Minigolf Adventures Deluxe',
 'playtime_forever': 25,
 'playtime_2weeks': 0},
{'item_id': '35010',
 'item_name': 'Batman: Arkham Asylum',
```

```
 'playtime_forever': 570,
 'playtime_2weeks': 0},
{'item_id': '35140',
 'item_name': 'Batman: Arkham Asylum GOTY Edition',
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '29180',
 'item_name': 'Osmos',
 'playtime_forever': 940,
 'playtime_2weeks': 0},
{'item_id': '15520',
 'item_name': 'AaAaAA!!! - A Reckless Disregard for Gravity',
 'playtime_forever': 110,
 'playtime_2weeks': 0},
{'item_id': '32370',
 'item_name': 'STAR WARS : Knights of the Old Republic ',
 'playtime_forever': 5,
 'playtime_2weeks': 0},
{'item_id': '37700',
 'item_name': 'Darkest of Days',
 'playtime_forever': 782,
 'playtime_2weeks': 0},
{'item_id': '6020',
 'item_name': 'STAR WARS  Jedi Knight: Jedi Academy ',
 'playtime_forever': 77,
 'playtime_2weeks': 0},
{'item_id': '24860',
 'item_name': 'Battlefield 2',
 'playtime_forever': 437,
 'playtime_2weeks': 0},
{'item_id': '39530',
 'item_name': 'Painkiller: Black Edition',
 'playtime_forever': 503,
 'playtime_2weeks': 0},
{'item_id': '550',
 'item_name': 'Left 4 Dead 2',
 'playtime_forever': 1474,
 'playtime_2weeks': 0},
{'item_id': '223530',
 'item_name': 'Left 4 Dead 2 Beta',
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '8980',
 'item_name': 'Borderlands',
 'playtime_forever': 3061,
 'playtime_2weeks': 0},
{'item_id': '41500',
```

```
 'item_name': 'Torchlight',
 'playtime_forever': 536,
 'playtime_2weeks': 0},
{'item_id': '20900',
 'item_name': 'The Witcher: Enhanced Edition',
 'playtime_forever': 139,
 'playtime_2weeks': 0},
{'item_id': '10180',
 'item_name': 'Call of Duty: Modern Warfare 2',
 'playtime_forever': 1886,
 'playtime_2weeks': 0},
{'item_id': '10190',
 'item_name': 'Call of Duty: Modern Warfare 2 - Multiplayer',
 'playtime_forever': 1784,
 'playtime_2weeks': 0},
{'item_id': '17450',
 'item_name': 'Dragon Age: Origins',
 'playtime_forever': 4431,
 'playtime_2weeks': 0},
{'item_id': '3170',
 'item_name': "King's Bounty: Armored Princess",
 'playtime_forever': 1021,
 'playtime_2weeks': 0},
{'item_id': '25900',
 'item_name': "King's Bounty: The Legend",
 'playtime_forever': 1389,
 'playtime_2weeks': 0},
{'item_id': '31410',
 'item_name': 'Zombie Driver',
 'playtime_forever': 95,
 'playtime_2weeks': 0},
{'item_id': '24980',
 'item_name': 'Mass Effect 2',
 'playtime_forever': 5001,
 'playtime_2weeks': 0},
{'item_id': '8850',
 'item_name': 'BioShock 2',
 'playtime_forever': 1504,
 'playtime_2weeks': 0},
{'item_id': '409720',
 'item_name': 'BioShock 2 Remastered',
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '46000',
 'item_name': 'Bob Came in Pieces',
 'playtime_forever': 11,
 'playtime_2weeks': 0},
```

```
{'item_id': '33230',
 'item_name': "Assassin's Creed II",
 'playtime_forever': 1636,
 'playtime_2weeks': 0},
{'item_id': '20820',
 'item_name': 'Shatter',
 'playtime_forever': 46,
 'playtime_2weeks': 0},
{'item_id': '47700',
 'item_name': 'Command and Conquer 4: Tiberian Twilight',
 'playtime_forever': 85,
 'playtime_2weeks': 0},
{'item_id': '24960',
 'item_name': 'Battlefield: Bad Company 2',
 'playtime_forever': 5716,
 'playtime_2weeks': 0},
{'item_id': '43110',
 'item_name': 'Metro 2033',
 'playtime_forever': 834,
 'playtime_2weeks': 0},
{'item_id': '8190',
 'item_name': 'Just Cause 2',
 'playtime_forever': 3083,
 'playtime_2weeks': 0},
{'item_id': '49600',
 'item_name': 'Beat Hazard',
 'playtime_forever': 127,
 'playtime_2weeks': 0},
{'item_id': '31220',
 'item_name': 'Sam & Max 301: The Penal Zone',
 'playtime_forever': 71,
 'playtime_2weeks': 0},
{'item_id': '31230',
 'item_name': 'Sam & Max 302: The Tomb of Sammun-Mak',
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '31240',
 'item_name': "Sam & Max 303: They Stole Max's Brain!",
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '31250',
 'item_name': 'Sam & Max 304: Beyond the Alley of the Dolls',
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '31260',
 'item_name': 'Sam & Max 305: The City that Dares not Sleep',
 'playtime_forever': 0,
```

```
 'playtime_2weeks': 0},
{'item_id': '21090',
 'item_name': 'F.E.A.R.',
 'playtime_forever': 585,
 'playtime_2weeks': 0},
{'item_id': '21110',
 'item_name': 'F.E.A.R.: Extraction Point',
 'playtime_forever': 146,
 'playtime_2weeks': 0},
{'item_id': '21120',
 'item_name': 'F.E.A.R.: Perseus Mandate',
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '49400',
 'item_name': 'Magic: The Gathering - Duels of the Planeswalkers',
 'playtime_forever': 32,
 'playtime_2weeks': 0},
{'item_id': '33910',
 'item_name': 'Arma 2',
 'playtime_forever': 3,
 'playtime_2weeks': 0},
{'item_id': '33930',
 'item_name': 'Arma 2: Operation Arrowhead',
 'playtime_forever': 3,
 'playtime_2weeks': 0},
{'item_id': '219540',
 'item_name': 'Arma 2: Operation Arrowhead Beta (Obsolete)',
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '17410',
 'item_name': "Mirror's Edge",
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '205790',
 'item_name': 'Dota 2 Test',
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '50130',
 'item_name': 'Mafia II',
 'playtime_forever': 707,
 'playtime_2weeks': 0},
{'item_id': '62100',
 'item_name': 'Chime',
 'playtime_forever': 50,
 'playtime_2weeks': 0},
{'item_id': '31280',
 'item_name': 'Poker Night at the Inventory',
```

```
 'playtime_forever': 206,
 'playtime_2weeks': 0},
{'item_id': '57300',
 'item_name': 'Amnesia: The Dark Descent',
 'playtime_forever': 12,
 'playtime_2weeks': 0},
{'item_id': '21970',
 'item_name': 'R.U.S.E',
 'playtime_forever': 64,
 'playtime_2weeks': 0},
{'item_id': '8930',
 'item_name': "Sid Meier's Civilization V",
 'playtime_forever': 10345,
 'playtime_2weeks': 0},
{'item_id': '41000',
 'item_name': 'Serious Sam HD: The First Encounter',
 'playtime_forever': 369,
 'playtime_2weeks': 0},
{'item_id': '41050',
 'item_name': 'Serious Sam Classic: The First Encounter',
 'playtime_forever': 76,
 'playtime_2weeks': 0},
{'item_id': '41060',
 'item_name': 'Serious Sam Classic: The Second Encounter',
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '227780',
 'item_name': 'Serious Sam Classics: Revolution',
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '18040',
 'item_name': 'DeathSpank',
 'playtime_forever': 594,
 'playtime_2weeks': 0},
{'item_id': '22380',
 'item_name': 'Fallout: New Vegas',
 'playtime_forever': 43,
 'playtime_2weeks': 0},
{'item_id': '42700',
 'item_name': 'Call of Duty: Black Ops',
 'playtime_forever': 966,
 'playtime_2weeks': 0},
{'item_id': '42710',
 'item_name': 'Call of Duty: Black Ops - Multiplayer',
 'playtime_forever': 2452,
 'playtime_2weeks': 0},
{'item_id': '62000',
```

```
 'item_name': 'Flight Control HD',
 'playtime_forever': 25,
 'playtime_2weeks': 0},
{'item_id': '40800',
 'item_name': 'Super Meat Boy',
 'playtime_forever': 119,
 'playtime_2weeks': 0},
{'item_id': '9500',
 'item_name': 'Gish',
 'playtime_forever': 6,
 'playtime_2weeks': 0},
{'item_id': '18700',
 'item_name': 'And Yet It Moves',
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '31270',
 'item_name': 'Puzzle Agent',
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '58200',
 'item_name': 'Jolly Rover',
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '70400',
 'item_name': "Recettear: An Item Shop's Tale",
 'playtime_forever': 537,
 'playtime_2weeks': 0},
{'item_id': '41300',
 'item_name': 'Altitude',
 'playtime_forever': 50,
 'playtime_2weeks': 0},
{'item_id': '41800',
 'item_name': 'Gratuitous Space Battles',
 'playtime_forever': 149,
 'playtime_2weeks': 0},
{'item_id': '42500',
 'item_name': 'DogFighter',
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '49900',
 'item_name': 'Plain Sight',
 'playtime_forever': 23,
 'playtime_2weeks': 0},
{'item_id': '55000',
 'item_name': 'Flotilla',
 'playtime_forever': 0,
 'playtime_2weeks': 0},
```

```
{'item_id': '34330',
 'item_name': 'Total War: SHOGUN 2',
 'playtime_forever': 1538,
 'playtime_2weeks': 0},
{'item_id': '63200',
 'item_name': 'Monday Night Combat',
 'playtime_forever': 573,
 'playtime_2weeks': 0},
{'item_id': '99810',
 'item_name': 'Bulletstorm',
 'playtime_forever': 2,
 'playtime_2weeks': 0},
{'item_id': '620',
 'item_name': 'Portal 2',
 'playtime_forever': 887,
 'playtime_2weeks': 0},
{'item_id': '42910',
 'item_name': 'Magicka',
 'playtime_forever': 588,
 'playtime_2weeks': 0},
{'item_id': '15500',
 'item_name': 'The Wonderful End of the World',
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '15540',
 'item_name': '1… 2… 3… KICK IT! (Drop That Beat Like an Ugly Baby)',
 'playtime_forever': 4,
 'playtime_2weeks': 0},
{'item_id': '18500',
 'item_name': 'Defense Grid: The Awakening',
 'playtime_forever': 126,
 'playtime_2weeks': 0},
{'item_id': '26500',
 'item_name': 'Cogs',
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '35460',
 'item_name': 'The Ball',
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '38700',
 'item_name': 'Toki Tori',
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '38720',
 'item_name': 'RUSH',
 'playtime_forever': 0,
```

```
 'playtime_2weeks': 0},
{'item_id': '63700',
 'item_name': 'BIT.TRIP BEAT',
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '280',
 'item_name': 'Half-Life: Source',
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '20920',
 'item_name': 'The Witcher 2: Assassins of Kings Enhanced Edition',
 'playtime_forever': 1343,
 'playtime_2weeks': 0},
{'item_id': '105600',
 'item_name': 'Terraria',
 'playtime_forever': 772,
 'playtime_2weeks': 0},
{'item_id': '98200',
 'item_name': 'Frozen Synapse',
 'playtime_forever': 9,
 'playtime_2weeks': 0},
{'item_id': '4500',
 'item_name': 'S.T.A.L.K.E.R.: Shadow of Chernobyl',
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '4540',
 'item_name': 'Titan Quest',
 'playtime_forever': 3,
 'playtime_2weeks': 0},
{'item_id': '4550',
 'item_name': 'Titan Quest: Immortal Throne',
 'playtime_forever': 76,
 'playtime_2weeks': 0},
{'item_id': '9480',
 'item_name': 'Saints Row 2',
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '15620',
 'item_name': 'Warhammer® 40,000 : Dawn of War® II',
 'playtime_forever': 2085,
 'playtime_2weeks': 0},
{'item_id': '20500',
 'item_name': 'Red Faction: Guerrilla Steam Edition',
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '20570',
 'item_name': 'Warhammer® 40,000 : Dawn of War® II - Chaos Rising ',
```

```
 'playtime_forever': 69,
 'playtime_2weeks': 0},
{'item_id': '50620',
 'item_name': 'Darksiders',
 'playtime_forever': 101,
 'playtime_2weeks': 0},
{'item_id': '55100',
 'item_name': 'Homefront',
 'playtime_forever': 51,
 'playtime_2weeks': 0},
{'item_id': '55140',
 'item_name': 'MX vs. ATV Reflex',
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '56400',
 'item_name': 'Warhammer® 40,000 : Dawn of War® II - Retribution ',
 'playtime_forever': 678,
 'playtime_2weeks': 0},
{'item_id': '475150',
 'item_name': 'Titan Quest Anniversary Edition',
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '96100',
 'item_name': 'Defy Gravity',
 'playtime_forever': 9,
 'playtime_2weeks': 0},
{'item_id': '35450',
 'item_name': 'Rising Storm/Red Orchestra 2 Multiplayer',
 'playtime_forever': 14194,
 'playtime_2weeks': 0},
{'item_id': '236830',
 'item_name': 'Red Orchestra 2: Heroes of Stalingrad - Single Player',
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '72850',
 'item_name': 'The Elder Scrolls V: Skyrim',
 'playtime_forever': 2512,
 'playtime_2weeks': 0},
{'item_id': '107100',
 'item_name': 'Bastion',
 'playtime_forever': 640,
 'playtime_2weeks': 0},
{'item_id': '57690',
 'item_name': 'Tropico 4',
 'playtime_forever': 1815,
 'playtime_2weeks': 0},
{'item_id': '91310',
```

```
 'item_name': 'Dead Island',
 'playtime_forever': 2078,
 'playtime_2weeks': 0},
{'item_id': '113200',
 'item_name': 'The Binding of Isaac',
 'playtime_forever': 1494,
 'playtime_2weeks': 0},
{'item_id': '65800',
 'item_name': 'Dungeon Defenders',
 'playtime_forever': 130,
 'playtime_2weeks': 0},
{'item_id': '9200',
 'item_name': 'RAGE',
 'playtime_forever': 653,
 'playtime_2weeks': 0},
{'item_id': '41070',
 'item_name': 'Serious Sam 3: BFE',
 'playtime_forever': 716,
 'playtime_2weeks': 0},
{'item_id': '48950',
 'item_name': 'Greed Corp',
 'playtime_forever': 72,
 'playtime_2weeks': 0},
{'item_id': '92800',
 'item_name': 'SpaceChem',
 'playtime_forever': 262,
 'playtime_2weeks': 0},
{'item_id': '104700',
 'item_name': 'Super Monday Night Combat',
 'playtime_forever': 44,
 'playtime_2weeks': 0},
{'item_id': '203770',
 'item_name': 'Crusader Kings II',
 'playtime_forever': 18,
 'playtime_2weeks': 0},
{'item_id': '203810',
 'item_name': 'Dear Esther',
 'playtime_forever': 86,
 'playtime_2weeks': 0},
{'item_id': '110800',
 'item_name': 'L.A. Noire',
 'playtime_forever': 1204,
 'playtime_2weeks': 0},
{'item_id': '207610',
 'item_name': 'The Walking Dead',
 'playtime_forever': 798,
 'playtime_2weeks': 0},
```

```
{'item_id': '98300',
 'item_name': 'Toy Soldiers',
 'playtime_forever': 110,
 'playtime_2weeks': 0},
{'item_id': '50300',
 'item_name': 'Spec Ops: The Line',
 'playtime_forever': 296,
 'playtime_2weeks': 0},
{'item_id': '204300',
 'item_name': 'Awesomenauts',
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '1700',
 'item_name': 'Arx Fatalis',
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '205100',
 'item_name': 'Dishonored',
 'playtime_forever': 1429,
 'playtime_2weeks': 0},
{'item_id': '211120',
 'item_name': 'The Political Machine',
 'playtime_forever': 1,
 'playtime_2weeks': 0},
{'item_id': '730',
 'item_name': 'Counter-Strike: Global Offensive',
 'playtime_forever': 23532,
 'playtime_2weeks': 0},
{'item_id': '49520',
 'item_name': 'Borderlands 2',
 'playtime_forever': 1303,
 'playtime_2weeks': 0},
{'item_id': '200260',
 'item_name': 'Batman: Arkham City GOTY',
 'playtime_forever': 100,
 'playtime_2weeks': 0},
{'item_id': '212680',
 'item_name': 'FTL: Faster Than Light',
 'playtime_forever': 500,
 'playtime_2weeks': 0},
{'item_id': '200710',
 'item_name': 'Torchlight II',
 'playtime_forever': 459,
 'playtime_2weeks': 0},
{'item_id': '219150',
 'item_name': 'Hotline Miami',
 'playtime_forever': 0,
```

```
 'playtime_2weeks': 0},
{'item_id': '215510',
 'item_name': 'Rocketbirds: Hardboiled Chicken',
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '8870',
 'item_name': 'BioShock Infinite',
 'playtime_forever': 2084,
 'playtime_2weeks': 0},
{'item_id': '200510',
 'item_name': 'XCOM: Enemy Unknown',
 'playtime_forever': 3478,
 'playtime_2weeks': 0},
{'item_id': '220240',
 'item_name': 'Far Cry® 3',
 'playtime_forever': 182,
 'playtime_2weeks': 0},
{'item_id': '218680',
 'item_name': 'Scribblenauts Unlimited',
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '228280',
 'item_name': "Baldur's Gate: Enhanced Edition",
 'playtime_forever': 905,
 'playtime_2weeks': 0},
{'item_id': '212070',
 'item_name': 'Star Conflict',
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '218230',
 'item_name': 'PlanetSide 2',
 'playtime_forever': 51,
 'playtime_2weeks': 0},
{'item_id': '236390',
 'item_name': 'War Thunder',
 'playtime_forever': 50,
 'playtime_2weeks': 0},
{'item_id': '3483',
 'item_name': 'Peggle Extreme',
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '203160',
 'item_name': 'Tomb Raider',
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '227080',
 'item_name': 'Papo & Yo',
```

```
 'playtime_forever': 81,
 'playtime_2weeks': 0},
{'item_id': '233450',
 'item_name': 'Prison Architect',
 'playtime_forever': 24,
 'playtime_2weeks': 0},
{'item_id': '43160',
 'item_name': 'Metro: Last Light',
 'playtime_forever': 712,
 'playtime_2weeks': 0},
{'item_id': '219740',
 'item_name': "Don't Starve",
 'playtime_forever': 139,
 'playtime_2weeks': 0},
{'item_id': '322330',
 'item_name': "Don't Starve Together",
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '222730',
 'item_name': 'Reus',
 'playtime_forever': 319,
 'playtime_2weeks': 0},
{'item_id': '241600',
 'item_name': 'Rogue Legacy',
 'playtime_forever': 73,
 'playtime_2weeks': 0},
{'item_id': '238930',
 'item_name': '7 Grand Steps, Step 1: What Ancients Begat',
 'playtime_forever': 159,
 'playtime_2weeks': 0},
{'item_id': '217140',
 'item_name': 'Rise of the Triad',
 'playtime_forever': 199,
 'playtime_2weeks': 0},
{'item_id': '39140',
 'item_name': 'FINAL FANTASY VII',
 'playtime_forever': 2896,
 'playtime_2weeks': 0},
{'item_id': '225080',
 'item_name': 'Brothers - A Tale of Two Sons',
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '239030',
 'item_name': 'Papers, Please',
 'playtime_forever': 194,
 'playtime_2weeks': 0},
{'item_id': '248820',
```

```
 'item_name': 'Risk of Rain',
 'playtime_forever': 272,
 'playtime_2weeks': 0},
{'item_id': '239350',
 'item_name': 'Spelunky',
 'playtime_forever': 458,
 'playtime_2weeks': 0},
{'item_id': '218620',
 'item_name': 'PAYDAY 2',
 'playtime_forever': 574,
 'playtime_2weeks': 0},
{'item_id': '206420',
 'item_name': 'Saints Row IV',
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '236850',
 'item_name': 'Europa Universalis IV',
 'playtime_forever': 89,
 'playtime_2weeks': 0},
{'item_id': '47790',
 'item_name': 'Medal of Honor(TM) Single Player',
 'playtime_forever': 230,
 'playtime_2weeks': 0},
{'item_id': '47830',
 'item_name': 'Medal of Honor(TM) Multiplayer',
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '250320',
 'item_name': 'The Wolf Among Us',
 'playtime_forever': 500,
 'playtime_2weeks': 0},
{'item_id': '250900',
 'item_name': 'The Binding of Isaac: Rebirth',
 'playtime_forever': 329,
 'playtime_2weeks': 0},
{'item_id': '204450',
 'item_name': 'Call of Juarez Gunslinger',
 'playtime_forever': 241,
 'playtime_2weeks': 0},
{'item_id': '233270',
 'item_name': 'Far Cry® 3 Blood Dragon',
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '251990',
 'item_name': 'Long Live The Queen',
 'playtime_forever': 135,
 'playtime_2weeks': 0},
```

```
  {'item_id': '253980',
   'item_name': 'Enclave',
   'playtime_forever': 0,
   'playtime_2weeks': 0},
  {'item_id': '236450',
   'item_name': 'PAC-MAN Championship Edition DX+',
   'playtime_forever': 154,
   'playtime_2weeks': 0},
  {'item_id': '242050',
   'item_name': "Assassin's Creed IV Black Flag",
   'playtime_forever': 1377,
   'playtime_2weeks': 0},
  {'item_id': '257350',
   'item_name': "Baldur's Gate II: Enhanced Edition",
   'playtime_forever': 0,
   'playtime_2weeks': 0},
  {'item_id': '213670',
   'item_name': 'South Park : The Stick of Truth ',
   'playtime_forever': 121,
   'playtime_2weeks': 0},
  {'item_id': '238010',
   'item_name': "Deus Ex: Human Revolution - Director's Cut",
   'playtime_forever': 2176,
   'playtime_2weeks': 0},
  {'item_id': '245470',
   'item_name': 'Democracy 3',
   'playtime_forever': 246,
   'playtime_2weeks': 0},
  {'item_id': '221910',
   'item_name': 'The Stanley Parable',
   'playtime_forever': 157,
   'playtime_2weeks': 0},
  {'item_id': '260230',
   'item_name': 'Valiant Hearts: The Great War  / Soldats Inconnus : Mémoires de
la Grande Guerre ',
   'playtime_forever': 467,
   'playtime_2weeks': 0},
  {'item_id': '238430',
   'item_name': 'Contagion',
   'playtime_forever': 77,
   'playtime_2weeks': 0},
  {'item_id': '261030',
   'item_name': 'The Walking Dead: Season Two',
   'playtime_forever': 567,
   'playtime_2weeks': 0},
  {'item_id': '261570',
   'item_name': 'Ori and the Blind Forest',
```

```
 'playtime_forever': 39,
 'playtime_2weeks': 0},
{'item_id': '263980',
 'item_name': 'Out There Somewhere',
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '39150',
 'item_name': 'FINAL FANTASY VIII',
 'playtime_forever': 15,
 'playtime_2weeks': 0},
{'item_id': '230230',
 'item_name': 'Divinity: Original Sin (Classic)',
 'playtime_forever': 522,
 'playtime_2weeks': 0},
{'item_id': '373420',
 'item_name': 'Divinity: Original Sin Enhanced Edition',
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '232790',
 'item_name': 'Broken Age',
 'playtime_forever': 216,
 'playtime_2weeks': 0},
{'item_id': '237990',
 'item_name': 'The Banner Saga',
 'playtime_forever': 19,
 'playtime_2weeks': 0},
{'item_id': '273620',
 'item_name': 'Freespace 2',
 'playtime_forever': 848,
 'playtime_2weeks': 0},
{'item_id': '222880',
 'item_name': 'Insurgency',
 'playtime_forever': 3031,
 'playtime_2weeks': 0},
{'item_id': '225840',
 'item_name': 'Sven Co-op',
 'playtime_forever': 26,
 'playtime_2weeks': 0},
{'item_id': '222900',
 'item_name': 'Dead Island: Epidemic',
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '280220',
 'item_name': 'Creeper World 3: Arc Eternal',
 'playtime_forever': 941,
 'playtime_2weeks': 0},
{'item_id': '236430',
```

```
 'item_name': 'DARK SOULS  II',
 'playtime_forever': 592,
 'playtime_2weeks': 0},
{'item_id': '201810',
 'item_name': 'Wolfenstein: The New Order',
 'playtime_forever': 599,
 'playtime_2weeks': 0},
{'item_id': '294860',
 'item_name': 'Valkyria Chronicles ',
 'playtime_forever': 1735,
 'playtime_2weeks': 0},
{'item_id': '295690',
 'item_name': 'PARTICLE MACE',
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '238460',
 'item_name': 'BattleBlock Theater',
 'playtime_forever': 407,
 'playtime_2weeks': 0},
{'item_id': '237930',
 'item_name': 'Transistor',
 'playtime_forever': 154,
 'playtime_2weeks': 0},
{'item_id': '301480',
 'item_name': 'GEARCRACK Arena',
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '303390',
 'item_name': 'Dead Bits',
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '306660',
 'item_name': 'Ultimate General: Gettysburg',
 'playtime_forever': 1245,
 'playtime_2weeks': 0},
{'item_id': '316790',
 'item_name': 'Grim Fandango Remastered',
 'playtime_forever': 42,
 'playtime_2weeks': 0},
{'item_id': '318430',
 'item_name': 'Squishy the Suicidal Pig',
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '241930',
 'item_name': 'Middle-earth: Shadow of Mordor',
 'playtime_forever': 889,
 'playtime_2weeks': 0},
```

```
{'item_id': '328080',
 'item_name': 'Retro-Pixel Castles',
 'playtime_forever': 5,
 'playtime_2weeks': 0},
{'item_id': '282070',
 'item_name': 'This War of Mine',
 'playtime_forever': 3,
 'playtime_2weeks': 0},
{'item_id': '333930',
 'item_name': 'Dirty Bomb',
 'playtime_forever': 2782,
 'playtime_2weeks': 0},
{'item_id': '334120',
 'item_name': 'Armikrog',
 'playtime_forever': 109,
 'playtime_2weeks': 0},
{'item_id': '257510',
 'item_name': 'The Talos Principle',
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '319630',
 'item_name': 'Life Is Strange ',
 'playtime_forever': 37,
 'playtime_2weeks': 0},
{'item_id': '346010',
 'item_name': 'Besiege',
 'playtime_forever': 464,
 'playtime_2weeks': 0},
{'item_id': '255710',
 'item_name': 'Cities: Skylines',
 'playtime_forever': 246,
 'playtime_2weeks': 0},
{'item_id': '232090',
 'item_name': 'Killing Floor 2',
 'playtime_forever': 6494,
 'playtime_2weeks': 0},
{'item_id': '232150',
 'item_name': 'Killing Floor 2 - SDK',
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '377160',
 'item_name': 'Fallout 4',
 'playtime_forever': 2001,
 'playtime_2weeks': 0},
{'item_id': '340170',
 'item_name': 'FINAL FANTASY TYPE-0 HD',
 'playtime_forever': 170,
```

```
 'playtime_2weeks': 0},
{'item_id': '391540',
 'item_name': 'Undertale',
 'playtime_forever': 395,
 'playtime_2weeks': 0},
{'item_id': '396660',
 'item_name': 'Tahira: Echoes of the Astral Empire',
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '337000',
 'item_name': 'Deus Ex: Mankind Divided ',
 'playtime_forever': 1159,
 'playtime_2weeks': 0},
{'item_id': '368230',
 'item_name': 'Kingdom: Classic',
 'playtime_forever': 83,
 'playtime_2weeks': 0},
{'item_id': '496300',
 'item_name': 'Kingdom: New Lands',
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '231430',
 'item_name': 'Company of Heroes 2',
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '289130',
 'item_name': 'Endless Legend',
 'playtime_forever': 593,
 'playtime_2weeks': 0},
{'item_id': '268500',
 'item_name': 'XCOM 2',
 'playtime_forever': 1495,
 'playtime_2weeks': 0},
{'item_id': '461640',
 'item_name': 'Sins Of The Demon RPG',
 'playtime_forever': 0,
 'playtime_2weeks': 0},
{'item_id': '379720',
 'item_name': 'DOOM',
 'playtime_forever': 793,
 'playtime_2weeks': 0},
{'item_id': '273350',
 'item_name': 'Evolve Stage 2',
 'playtime_forever': 58,
 'playtime_2weeks': 0}]}
```

```
[ ]: # Merger Line with comma & Build List
merge_data = '[' + ','.join(lines) + ']'
```

```
[ ]: data_linter = ast.literal_eval(merge_data)
```

```
[ ]: # Export to JSON

with open('data.json', 'w') as j_file:
    json.dump(data_linter, j_file)
```

We now have a `.json` file that we can easily view as a Pandas DataFrame.

```
[ ]: data_frame = pd.DataFrame(data_linter)
data_frame.head()
```

```
[ ]:              user_id  items_count           steam_id  \
0  76561197970982479          277  76561197970982479
1            js41637          888  76561198035864385
2          evcentric          137  76561198007712555
3         Riot-Punch          328  76561197963445855
4              doctr          541  76561198002099482

                                           user_url  \
0  http://steamcommunity.com/profiles/76561197970…
1               http://steamcommunity.com/id/js41637
2             http://steamcommunity.com/id/evcentric
3            http://steamcommunity.com/id/Riot-Punch
4                 http://steamcommunity.com/id/doctr

                                               items
0  [{'item_id': '10', 'item_name': 'Counter-Strik…
1  [{'item_id': '10', 'item_name': 'Counter-Strik…
2  [{'item_id': '1200', 'item_name': 'Red Orchest…
3  [{'item_id': '10', 'item_name': 'Counter-Strik…
4  [{'item_id': '300', 'item_name': 'Day of Defea…
```

1.2 Items detail data

This dataset was downloaded from https://cseweb.ucsd.edu/~jmcauley/datasets.html#steam_data file *Version 2: item metadata*

```
[ ]: # Load data

with open("C:
↪\\Users\\pmogh\\Documents\\Python_Workspace\\Final_Project_Dataset\\ProjectData\\Part_2\\st
↪json",encoding="utf-8") as game_details:
    data_game = game_details.readlines()
```

[ ]:
```
# View first line
data_game[0]
```

[ ]:
```
"{u'publisher': u'Kotoshiro', u'genres': [u'Action', u'Casual', u'Indie',
u'Simulation', u'Strategy'], u'app_name': u'Lost Summoner Kitty', u'title':
u'Lost Summoner Kitty', u'url':
u'http://store.steampowered.com/app/761140/Lost_Summoner_Kitty/',
u'release_date': u'2018-01-04', u'tags': [u'Strategy', u'Action', u'Indie',
u'Casual', u'Simulation'], u'discount_price': 4.49, u'reviews_url':
u'http://steamcommunity.com/app/761140/reviews/?browsefilter=mostrecent&p=1',
u'specs': [u'Single-player'], u'price': 4.99, u'early_access': False, u'id':
u'761140', u'developer': u'Kotoshiro'}\n"
```

[ ]:
```
# Get number of lines
len(data_game)
```

[ ]:
```
32135
```

There are 32135 lines, each representing a different game.

[ ]:
```
# evaluate the first string
data_linter = ast.literal_eval(data_game[0])
```

[ ]:
```
# View the first line
data_linter
```

[ ]:
```
{'publisher': 'Kotoshiro',
 'genres': ['Action', 'Casual', 'Indie', 'Simulation', 'Strategy'],
 'app_name': 'Lost Summoner Kitty',
 'title': 'Lost Summoner Kitty',
 'url': 'http://store.steampowered.com/app/761140/Lost_Summoner_Kitty/',
 'release_date': '2018-01-04',
 'tags': ['Strategy', 'Action', 'Indie', 'Casual', 'Simulation'],
 'discount_price': 4.49,
 'reviews_url':
'http://steamcommunity.com/app/761140/reviews/?browsefilter=mostrecent&p=1',
 'specs': ['Single-player'],
 'price': 4.99,
 'early_access': False,
 'id': '761140',
 'developer': 'Kotoshiro'}
```

[ ]:
```
merge_data = '[' + ','.join(data_game) + ']'
```

[ ]:
```
data_linter = ast.literal_eval(merge_data)
```

[ ]:
```
with open('gamesdata.json', 'w') as game_data:
    json.dump(data_linter, game_data)
```

We now have a `.json` file that we can easily view as a Pandas DataFrame.

[ ]:
```
data_frame = pd.DataFrame(data_linter)
data_frame.head()
```

[ ]:
```
          publisher                                              genres  \
0        Kotoshiro      [Action, Casual, Indie, Simulation, Strategy]
1  Making Fun, Inc.                [Free to Play, Indie, RPG, Strategy]
2     Poolians.com  [Casual, Free to Play, Indie, Simulation, Sports]
3                                        [Action, Adventure, Casual]
4              NaN                                                NaN

                 app_name                    title  \
0     Lost Summoner Kitty      Lost Summoner Kitty
1               Ironbound                Ironbound
2  Real Pool 3D - Poolians  Real Pool 3D - Poolians
3                    2222                    2222
4           Log Challenge                      NaN

                                             url release_date  \
0  http://store.steampowered.com/app/761140/Lost_…   2018-01-04
1  http://store.steampowered.com/app/643980/Ironb…   2018-01-04
2  http://store.steampowered.com/app/670290/Real_…   2017-07-24
3    http://store.steampowered.com/app/767400/2222/   2017-12-07
4  http://store.steampowered.com/app/773570/Log_C…          NaN

                                              tags  discount_price  \
0     [Strategy, Action, Indie, Casual, Simulation]            4.49
1  [Free to Play, Strategy, Indie, RPG, Card Game…             NaN
2  [Free to Play, Simulation, Sports, Casual, Ind…             NaN
3                      [Action, Adventure, Casual]            0.83
4                   [Action, Indie, Casual, Sports]            1.79

                                       reviews_url  \
0  http://steamcommunity.com/app/761140/reviews/?…
1  http://steamcommunity.com/app/643980/reviews/?…
2  http://steamcommunity.com/app/670290/reviews/?…
3  http://steamcommunity.com/app/767400/reviews/?…
4  http://steamcommunity.com/app/773570/reviews/?…

                                             specs         price  \
0                                  [Single-player]          4.99
1  [Single-player, Multi-player, Online Multi-Pla…  Free To Play
2  [Single-player, Multi-player, Online Multi-Pla…  Free to Play
3                                  [Single-player]          0.99
4  [Single-player, Full controller support, HTC V…          2.99
```

	early_access	id	developer	sentiment	metascore
0	False	761140	Kotoshiro	NaN	NaN
1	False	643980	Secret Level SRL	Mostly Positive	NaN
2	False	670290	Poolians.com	Mostly Positive	NaN
3	False	767400		NaN	NaN
4	False	773570	NaN	NaN	NaN

Part-2-Data-Preprocessing

December 21, 2021

1 Data Preprocessing

In this process, we load both the dataset and extract the user id and game id.After this we combine the data.

[ ]:
```
# Import libraries
import pandas as pd
import numpy as np

#Import Warnings
import warnings
warnings.filterwarnings("ignore")
```

1.1 Games data

Loading the game datafile for preprocessing.Evey row in data is different game.

[ ]:
```
game_data_frame = pd.read_json('gamesdata.json')
game_data_frame.head()
```

[ ]:
```
          publisher                                              genres  \
0         Kotoshiro      [Action, Casual, Indie, Simulation, Strategy]
1  Making Fun, Inc.                [Free to Play, Indie, RPG, Strategy]
2      Poolians.com  [Casual, Free to Play, Indie, Simulation, Sports]
3                                       [Action, Adventure, Casual]
4               NaN                                                NaN

                  app_name                     title  \
0     Lost Summoner Kitty      Lost Summoner Kitty
1                Ironbound                Ironbound
2  Real Pool 3D - Poolians  Real Pool 3D - Poolians
3                     2222                     2222
4           Log Challenge                      NaN

                                                 url release_date  \
0  http://store.steampowered.com/app/761140/Lost_…   2018-01-04
1  http://store.steampowered.com/app/643980/Ironb…   2018-01-04
2  http://store.steampowered.com/app/670290/Real_…   2017-07-24
```

```
3     http://store.steampowered.com/app/767400/2222/   2017-12-07
4  http://store.steampowered.com/app/773570/Log_C…          NaN

                                                 tags  discount_price  \
0      [Strategy, Action, Indie, Casual, Simulation]            4.49
1  [Free to Play, Strategy, Indie, RPG, Card Game…             NaN
2  [Free to Play, Simulation, Sports, Casual, Ind…             NaN
3                        [Action, Adventure, Casual]            0.83
4                    [Action, Indie, Casual, Sports]            1.79

                                        reviews_url  \
0  http://steamcommunity.com/app/761140/reviews/?…
1  http://steamcommunity.com/app/643980/reviews/?…
2  http://steamcommunity.com/app/670290/reviews/?…
3  http://steamcommunity.com/app/767400/reviews/?…
4  http://steamcommunity.com/app/773570/reviews/?…

                                             specs         price  \
0                                   [Single-player]          4.99
1  [Single-player, Multi-player, Online Multi-Pla…  Free To Play
2  [Single-player, Multi-player, Online Multi-Pla…  Free to Play
3                                   [Single-player]          0.99
4  [Single-player, Full controller support, HTC V…          2.99

   early_access        id         developer        sentiment metascore
0         False  761140.0         Kotoshiro              NaN       NaN
1         False  643980.0  Secret Level SRL  Mostly Positive       NaN
2         False  670290.0      Poolians.com  Mostly Positive       NaN
3         False  767400.0                               NaN       NaN
4         False  773570.0               NaN              NaN       NaN
```

We note that there are certain features which are lists, namely genres, tags and specs. These will be investigated further in the Data Exploration phase.

[ ]:
```
# Save as csv
game_data_frame.to_csv('gamesdata.csv')
```

1.2 User items Data

We now load the user items data, which has users as rows and details regarding items owned as columns.

[ ]:
```
# Load users/items data
user_data_frame = pd.read_json('data.json')
user_data_frame.head()
```

[ ]:
```
             user_id  items_count           steam_id  \
0  76561197970982479          277  76561197970982480
```

```
1            js41637          888  76561198035864384
2          evcentric          137  76561198007712560
3         Riot-Punch          328  76561197963445856
4              doctr          541  76561198002099488

                                                user_url  \
0  http://steamcommunity.com/profiles/76561197970…
1               http://steamcommunity.com/id/js41637
2             http://steamcommunity.com/id/evcentric
3            http://steamcommunity.com/id/Riot-Punch
4                 http://steamcommunity.com/id/doctr

                                                   items
0  [{'item_id': '10', 'item_name': 'Counter-Strik…
1  [{'item_id': '10', 'item_name': 'Counter-Strik…
2  [{'item_id': '1200', 'item_name': 'Red Orchest…
3  [{'item_id': '10', 'item_name': 'Counter-Strik…
4  [{'item_id': '300', 'item_name': 'Day of Defea…
```

We break the user and item data into seprate groups

[ ]:
```
# Extract items_count feat
numgames = user_data_frame[['user_id', 'items_count']]
numgames.head()
```

[ ]:
```
             user_id  items_count
0  76561197970982479          277
1            js41637          888
2          evcentric          137
3         Riot-Punch          328
4              doctr          541
```

[ ]:
```
# Save as csv
numgames.to_csv('numgames.csv')
```

We note that the `items` column appears to be a list of dictionaries. Let us look at it in further detail.

[ ]:
```
# Preview items column values for first user
# Restrict to first 2 items in dictionary

user_data_frame['items'][0][0:2]
```

[ ]:
```
[{'item_id': '10',
  'item_name': 'Counter-Strike',
  'playtime_forever': 6,
  'playtime_2weeks': 0},
 {'item_id': '20',
```

```
'item_name': 'Team Fortress Classic',
'playtime_forever': 0,
'playtime_2weeks': 0}]
```

Each game is represented by a dictionary with keys the game's `item_id`, `item_name`, `playtime_forever` and `playtime_2weeks`. The dictionaries are then storred in a list.

We will look to extract the `item_id`s into a seperate column. For now we will leave the playtime data but look to incorporate it later.

[ ]:
```
# Get all item_id for first user
gameids = [user_data_frame['items'][0][index]['item_id'] for index, _ in␣
↪enumerate(user_data_frame['items'][0])]
# Show first 10 item ids
gameids[:10]
```

[ ]:
```
['10', '20', '30', '40', '50', '60', '70', '130', '300', '240']
```

We will now generalize the above and create a column extracting the `item_id` from the dictionaries for each user.

[ ]:
```
# Create column with item ids
user_data_frame['item_id'] = user_data_frame['items'].apply(lambda x: [x␣
↪[index]['item_id'] for index, _ in enumerate(x)])
user_data_frame.head()
```

[ ]:
```
              user_id  items_count           steam_id  \
0  76561197970982479          277  76561197970982480
1            js41637          888  76561198035864384
2          evcentric          137  76561198007712560
3         Riot-Punch          328  76561197963445856
4              doctr          541  76561198002099488

                                           user_url  \
0  http://steamcommunity.com/profiles/76561197970…
1              http://steamcommunity.com/id/js41637
2            http://steamcommunity.com/id/evcentric
3           http://steamcommunity.com/id/Riot-Punch
4                http://steamcommunity.com/id/doctr

                                               items  \
0  [{'item_id': '10', 'item_name': 'Counter-Strik…
1  [{'item_id': '10', 'item_name': 'Counter-Strik…
2  [{'item_id': '1200', 'item_name': 'Red Orchest…
3  [{'item_id': '10', 'item_name': 'Counter-Strik…
4  [{'item_id': '300', 'item_name': 'Day of Defea…

                                             item_id
0  [10, 20, 30, 40, 50, 60, 70, 130, 300, 240, 38…
```

```
1  [10, 80, 100, 300, 30, 40, 60, 240, 280, 360, …
2  [1200, 1230, 1280, 1520, 220, 320, 340, 360, 3…
3  [10, 20, 30, 40, 50, 60, 70, 130, 80, 100, 300…
4  [300, 20, 50, 70, 130, 10, 30, 40, 60, 80, 100…
```

As the unique user `steam_id` is a large integer, we will replace it with a new `uid` counter, which starts at 0 and increments by 1 (like the index).

We will also only select the relevant columns for the purpose of building a user-item interactions matrix, namely the newly created user id `iud` and the `item_id`.

[ ]:
```
# Add a column with substitute user_id, counter
user_data_frame['uid'] = np.arange(len(user_data_frame))

# Take relevant columns
useritems = user_data_frame[['uid', 'item_id']]

# Check
useritems.head()
```

[ ]:
```
   uid                                             item_id
0    0  [10, 20, 30, 40, 50, 60, 70, 130, 300, 240, 38…
1    1  [10, 80, 100, 300, 30, 40, 60, 240, 280, 360, …
2    2  [1200, 1230, 1280, 1520, 220, 320, 340, 360, 3…
3    3  [10, 20, 30, 40, 50, 60, 70, 130, 80, 100, 300…
4    4  [300, 20, 50, 70, 130, 10, 30, 40, 60, 80, 100…
```

The next step is to explode the `item_id` into seperate rows, so each user-item interaction has it's own row.

[ ]:
```
# Explode item_ids into seperate rows
lst_col = 'item_id'
useritems = pd.DataFrame({col:np.repeat(useritems[col].values,␣
 ↪useritems[lst_col].str.len())
                                for col in useritems.columns.difference([lst_col])
                            }).assign(**{lst_col:np.concatenate(useritems[lst_col].
 ↪values)})[useritems.columns.tolist()]
useritems
```

[ ]:
```
          uid item_id
0           0      10
1           0      20
2           0      30
3           0      40
4           0      50
…         …      …
994981   9999    7670
994982   9999    8850
994983   9999    8870
```

```
994984  9999  409710
994985  9999  409720

[994986 rows x 2 columns]
```

As we are concerned with whether the game is owned, as opposed to ratings, we will add a binary column `owned` which will have 1s everywhere, as only items owned appear in the DataFrame.

[ ]:
```
# Add binary owned column
useritems['owned'] = np.ones(shape = useritems.shape[0])

# Check
useritems.head()
```

[ ]:
```
   uid item_id  owned
0    0      10    1.0
1    0      20    1.0
2    0      30    1.0
3    0      40    1.0
4    0      50    1.0
```

[ ]:
```
len(useritems)
```

[ ]:
```
994986
```

We note that we have over 5 million individual user-item relationships represented in our DataFrame.

We want to restrict ourselves to user-item relationships where the item is in the first `gamesdata` DataFrame to be able to extract relevant information such as genre.

We will ensure that the DataFrames can be merged on the game `id` feature by changing the type and column name.

[ ]:
```
# Change item_id to int
useritems['item_id'] = useritems['item_id'].astype(int)

# Rename column to match
useritems = useritems.rename(columns={'item_id': 'id'})
```

We can now merge the DataFrames.

[ ]:
```
# Merge useritems and games data dataframes
alldata = pd.merge(useritems, game_data_frame, on = 'id')
alldata.head()
```

[ ]:
```
   uid  id  owned publisher    genres        app_name           title  \
0    0  10    1.0     Valve  [Action]  Counter-Strike  Counter-Strike
1    1  10    1.0     Valve  [Action]  Counter-Strike  Counter-Strike
2    3  10    1.0     Valve  [Action]  Counter-Strike  Counter-Strike
```

```
3    4  10    1.0     Valve  [Action]  Counter-Strike  Counter-Strike
4   10  10    1.0     Valve  [Action]  Counter-Strike  Counter-Strike

                                                 url release_date  \
0  http://store.steampowered.com/app/10/CounterSt…   2000-11-01
1  http://store.steampowered.com/app/10/CounterSt…   2000-11-01
2  http://store.steampowered.com/app/10/CounterSt…   2000-11-01
3  http://store.steampowered.com/app/10/CounterSt…   2000-11-01
4  http://store.steampowered.com/app/10/CounterSt…   2000-11-01

                                               tags  discount_price  \
0  [Action, FPS, Multiplayer, Shooter, Classic, T…             NaN
1  [Action, FPS, Multiplayer, Shooter, Classic, T…             NaN
2  [Action, FPS, Multiplayer, Shooter, Classic, T…             NaN
3  [Action, FPS, Multiplayer, Shooter, Classic, T…             NaN
4  [Action, FPS, Multiplayer, Shooter, Classic, T…             NaN

                                        reviews_url  \
0  http://steamcommunity.com/app/10/reviews/?brow…
1  http://steamcommunity.com/app/10/reviews/?brow…
2  http://steamcommunity.com/app/10/reviews/?brow…
3  http://steamcommunity.com/app/10/reviews/?brow…
4  http://steamcommunity.com/app/10/reviews/?brow…

                                    specs price  early_access developer  \
0  [Multi-player, Valve Anti-Cheat enabled]  9.99         False     Valve
1  [Multi-player, Valve Anti-Cheat enabled]  9.99         False     Valve
2  [Multi-player, Valve Anti-Cheat enabled]  9.99         False     Valve
3  [Multi-player, Valve Anti-Cheat enabled]  9.99         False     Valve
4  [Multi-player, Valve Anti-Cheat enabled]  9.99         False     Valve

                sentiment metascore
0  Overwhelmingly Positive        88
1  Overwhelmingly Positive        88
2  Overwhelmingly Positive        88
3  Overwhelmingly Positive        88
4  Overwhelmingly Positive        88
```

[ ]:
```
def find_total_perc_missing (data_set):
    temp_missing_val = (data_set.isnull().sum()).sum()
    total_cel = np.product(data_set.shape)
    perc_missing_data=100 * (temp_missing_val/total_cel)
    return perc_missing_data
```

[ ]:
```
find_total_perc_missing(alldata)
```

[ ]:
```
8.517928035516288
```

[ ]:
```
def find_missing_value(data_set):
    percent_missing = data_set.isnull().sum() * 100 / len(data_set)
    missing_value_df = pd.DataFrame({'column_name': data_set.
 ↪columns,'percent_missing': percent_missing})
    missing_value_df=missing_value_df.sort_values('percent_missing',␣
 ↪ascending=False)

    return missing_value_df
```

[ ]:
```
find_missing_value(alldata).head(15)
```

[ ]:
```
                   column_name  percent_missing
discount_price  discount_price        99.872911
metascore            metascore        39.161547
publisher            publisher         3.329353
developer            developer         2.629344
genres                  genres         2.231307
release_date      release_date         1.914243
price                    price         1.908853
title                    title         1.779847
specs                    specs         0.450382
sentiment            sentiment         0.042882
tags                      tags         0.002036
reviews_url        reviews_url         0.000000
id                          id         0.000000
url                        url         0.000000
early_access      early_access         0.000000
```

[ ]:
```
alldata.shape
```

[ ]:
```
(834847, 18)
```

[ ]:
```
len(alldata)
```

[ ]:
```
834847
```

We have more than 8 lakh data

[ ]:
```
# Drop entries with no title
datawithnames = alldata.dropna(axis=0, subset=['title'])
datawithnames.head()
```

[ ]:
```
    uid  id  owned publisher    genres        app_name           title  \
0     0  10    1.0     Valve  [Action]  Counter-Strike  Counter-Strike
1     1  10    1.0     Valve  [Action]  Counter-Strike  Counter-Strike
2     3  10    1.0     Valve  [Action]  Counter-Strike  Counter-Strike
3     4  10    1.0     Valve  [Action]  Counter-Strike  Counter-Strike
4    10  10    1.0     Valve  [Action]  Counter-Strike  Counter-Strike
```

```
                                                 url release_date  \
0  http://store.steampowered.com/app/10/CounterSt…   2000-11-01
1  http://store.steampowered.com/app/10/CounterSt…   2000-11-01
2  http://store.steampowered.com/app/10/CounterSt…   2000-11-01
3  http://store.steampowered.com/app/10/CounterSt…   2000-11-01
4  http://store.steampowered.com/app/10/CounterSt…   2000-11-01

                                               tags  discount_price  \
0  [Action, FPS, Multiplayer, Shooter, Classic, T…             NaN
1  [Action, FPS, Multiplayer, Shooter, Classic, T…             NaN
2  [Action, FPS, Multiplayer, Shooter, Classic, T…             NaN
3  [Action, FPS, Multiplayer, Shooter, Classic, T…             NaN
4  [Action, FPS, Multiplayer, Shooter, Classic, T…             NaN

                                         reviews_url  \
0  http://steamcommunity.com/app/10/reviews/?brow…
1  http://steamcommunity.com/app/10/reviews/?brow…
2  http://steamcommunity.com/app/10/reviews/?brow…
3  http://steamcommunity.com/app/10/reviews/?brow…
4  http://steamcommunity.com/app/10/reviews/?brow…

                                    specs price  early_access developer  \
0  [Multi-player, Valve Anti-Cheat enabled]  9.99         False     Valve
1  [Multi-player, Valve Anti-Cheat enabled]  9.99         False     Valve
2  [Multi-player, Valve Anti-Cheat enabled]  9.99         False     Valve
3  [Multi-player, Valve Anti-Cheat enabled]  9.99         False     Valve
4  [Multi-player, Valve Anti-Cheat enabled]  9.99         False     Valve

                 sentiment metascore
0  Overwhelmingly Positive        88
1  Overwhelmingly Positive        88
2  Overwhelmingly Positive        88
3  Overwhelmingly Positive        88
4  Overwhelmingly Positive        88
```

[ ]:
```
len(datawithnames)
```

[ ]:
```
819988
```

We will save this DataFrame as a `csv` file to conduct data exploration and gain insights.

[ ]:
```
# Save to csv
datawithnames.to_csv('mergeddata.csv')
```

Finally, let us extract the relevant columns for our user-item interactions matrix.

[ ]:
```
# Get relevant columns for recommendation engine
recdata = datawithnames[['uid','id','owned']]
recdata.head()
```

[ ]:

	uid	id	owned
0	0	10	1.0
1	1	10	1.0
2	3	10	1.0
3	4	10	1.0
4	10	10	1.0

[ ]:
```
# Save to csv
recdata.to_csv('recdata.csv')
```

Part-3-Data-Operation

December 21, 2021

1 Data Operations

Using the above data CSV file, we now gain insights.

[ ]:
```
# Import libraries
import pandas as pd
import numpy as np

import pickle

import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

from collections import Counter

#Import Warnings
import warnings
warnings.filterwarnings("ignore")
```

1.1 Load Data

loading the game data with all the its columns.

[ ]:
```
# Load games data
gamesdata = pd.read_csv('gamesdata.csv', index_col = 0)
gamesdata.head()
```

[ ]:
```
          publisher                                              genres  \
0         Kotoshiro  ['Action', 'Casual', 'Indie', 'Simulation', 'S…
1  Making Fun, Inc.       ['Free to Play', 'Indie', 'RPG', 'Strategy']
2     Poolians.com  ['Casual', 'Free to Play', 'Indie', 'Simulatio…
3                                ['Action', 'Adventure', 'Casual']
4               NaN                                                NaN

                  app_name                    title  \
0      Lost Summoner Kitty      Lost Summoner Kitty
1                Ironbound                Ironbound
```

```
2  Real Pool 3D - Poolians  Real Pool 3D - Poolians
3                     2222                     2222
4             Log Challenge                      NaN

                                                 url release_date  \
0  http://store.steampowered.com/app/761140/Lost_…   2018-01-04
1  http://store.steampowered.com/app/643980/Ironb…   2018-01-04
2  http://store.steampowered.com/app/670290/Real_…   2017-07-24
3     http://store.steampowered.com/app/767400/2222/   2017-12-07
4  http://store.steampowered.com/app/773570/Log_C…          NaN

                                               tags  discount_price  \
0  ['Strategy', 'Action', 'Indie', 'Casual', 'Sim…            4.49
1  ['Free to Play', 'Strategy', 'Indie', 'RPG', '…             NaN
2  ['Free to Play', 'Simulation', 'Sports', 'Casu…             NaN
3                  ['Action', 'Adventure', 'Casual']            0.83
4            ['Action', 'Indie', 'Casual', 'Sports']            1.79

                                          reviews_url  \
0  http://steamcommunity.com/app/761140/reviews/?…
1  http://steamcommunity.com/app/643980/reviews/?…
2  http://steamcommunity.com/app/670290/reviews/?…
3  http://steamcommunity.com/app/767400/reviews/?…
4  http://steamcommunity.com/app/773570/reviews/?…

                                              specs         price  \
0                                ['Single-player']          4.99
1  ['Single-player', 'Multi-player', 'Online Mult…  Free To Play
2  ['Single-player', 'Multi-player', 'Online Mult…  Free to Play
3                                ['Single-player']          0.99
4  ['Single-player', 'Full controller support', '…          2.99

   early_access        id         developer        sentiment  metascore
0         False  761140.0         Kotoshiro              NaN        NaN
1         False  643980.0  Secret Level SRL  Mostly Positive        NaN
2         False  670290.0      Poolians.com  Mostly Positive        NaN
3         False  767400.0                               NaN        NaN
4         False  773570.0               NaN              NaN        NaN
```

We also load the `mergeddata.csv` file which has a row for each user-item interaction.

[ ]:
```
# Load merged data
mergeddata = pd.read_csv('mergeddata.csv', index_col = 0)
mergeddata.head()
```

[ ]:
```
   uid  id  owned publisher      genres        app_name           title  \
0    0  10    1.0     Valve  ['Action']  Counter-Strike  Counter-Strike
1    1  10    1.0     Valve  ['Action']  Counter-Strike  Counter-Strike
```

```
2    3  10    1.0     Valve  ['Action']  Counter-Strike  Counter-Strike
3    4  10    1.0     Valve  ['Action']  Counter-Strike  Counter-Strike
4   10  10    1.0     Valve  ['Action']  Counter-Strike  Counter-Strike

                                                 url release_date  \
0  http://store.steampowered.com/app/10/CounterSt…   2000-11-01
1  http://store.steampowered.com/app/10/CounterSt…   2000-11-01
2  http://store.steampowered.com/app/10/CounterSt…   2000-11-01
3  http://store.steampowered.com/app/10/CounterSt…   2000-11-01
4  http://store.steampowered.com/app/10/CounterSt…   2000-11-01

                                                tags  discount_price  \
0  ['Action', 'FPS', 'Multiplayer', 'Shooter', 'C…             NaN
1  ['Action', 'FPS', 'Multiplayer', 'Shooter', 'C…             NaN
2  ['Action', 'FPS', 'Multiplayer', 'Shooter', 'C…             NaN
3  ['Action', 'FPS', 'Multiplayer', 'Shooter', 'C…             NaN
4  ['Action', 'FPS', 'Multiplayer', 'Shooter', 'C…             NaN

                                          reviews_url  \
0  http://steamcommunity.com/app/10/reviews/?brow…
1  http://steamcommunity.com/app/10/reviews/?brow…
2  http://steamcommunity.com/app/10/reviews/?brow…
3  http://steamcommunity.com/app/10/reviews/?brow…
4  http://steamcommunity.com/app/10/reviews/?brow…

                                       specs price  early_access developer  \
0  ['Multi-player', 'Valve Anti-Cheat enabled']  9.99         False     Valve
1  ['Multi-player', 'Valve Anti-Cheat enabled']  9.99         False     Valve
2  ['Multi-player', 'Valve Anti-Cheat enabled']  9.99         False     Valve
3  ['Multi-player', 'Valve Anti-Cheat enabled']  9.99         False     Valve
4  ['Multi-player', 'Valve Anti-Cheat enabled']  9.99         False     Valve

                 sentiment  metascore
0  Overwhelmingly Positive       88.0
1  Overwhelmingly Positive       88.0
2  Overwhelmingly Positive       88.0
3  Overwhelmingly Positive       88.0
4  Overwhelmingly Positive       88.0
```

And finally we load the `numgames.csv` file which just lists the number of games owned for each user.

[ ]:
```
# Load numgames data
numgames = pd.read_csv('numgames.csv', index_col = 0)
numgames.head()
```

[ ]:
```
              user_id  items_count
0  76561197970982479          277
1             js41637          888
2           evcentric          137
3         Riot-Punch          328
4              doctr          541
```

1.2 Exploration

1.2.1 User interaction data

[ ]:
```
mergeddata['id'].nunique()
```

[ ]:
```
8171
```

[ ]:
```
mergeddata['uid'].nunique()
```

[ ]:
```
8769
```

1.2.2 Release date

[ ]:
```
# Select entries where release date is not null
data = gamesdata[gamesdata['release_date'].notnull()]
```

[ ]:
```
# Describe feature
data['release_date'].describe()
```

[ ]:
```
count          30068
unique          3582
top       2012-10-16
freq             100
Name: release_date, dtype: object
```

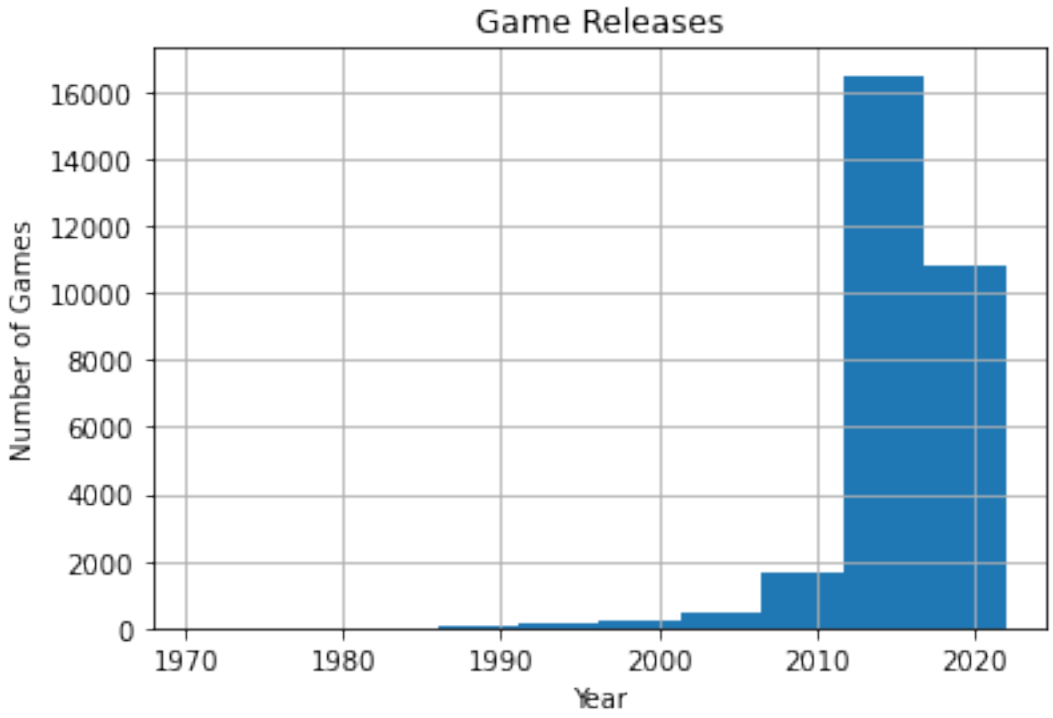
We note that there are 3582 unique values. We want to convert the type to Datetime instead of object.

[ ]:
```
# Replace strings which are not of the format xxxx-xx-xx with None
data['release_date'] = data['release_date'].map(lambda x : x if x[-3] ==␣
 ↪'-'else None)

# Select entries where release date is not null
data = data[data['release_date'].notnull()]

# Convert to DateTime
data['release_date'] = pd.to_datetime(data['release_date'])

# Check
data['release_date'].describe()
```

[ ]:
```
count                   29783
unique                   3457
top       2012-10-16 00:00:00
freq                      100
first     1970-07-15 00:00:00
last      2021-12-31 00:00:00
Name: release_date, dtype: object
```

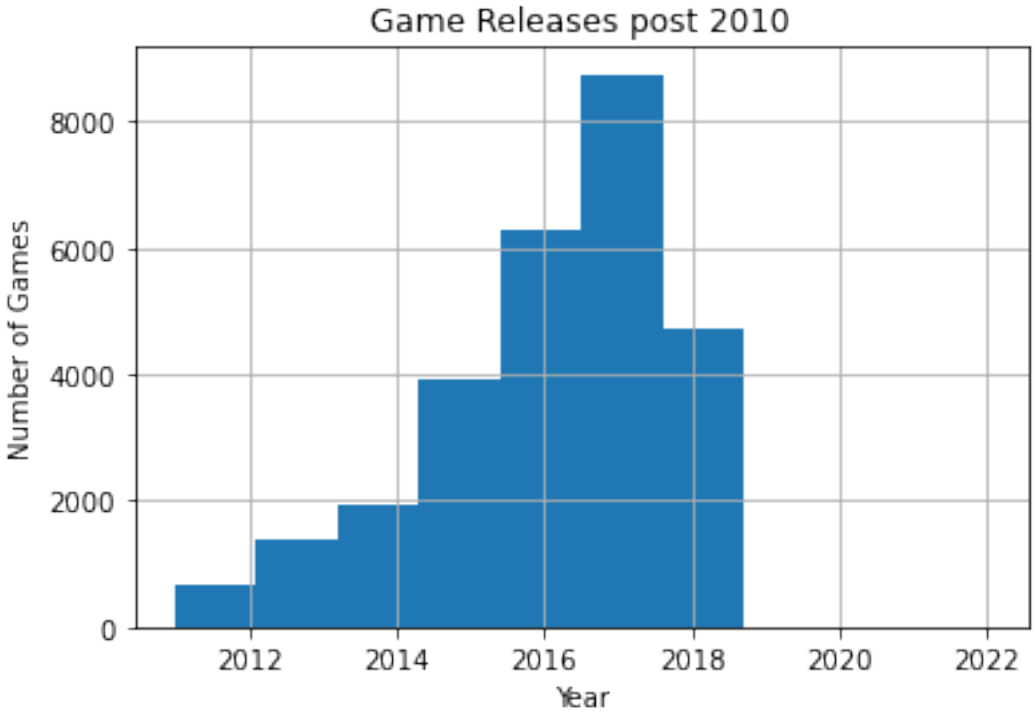
We see that our data contains games ranging from 1970 up to predicted release date of December 2021.

[ ]:
```
# Plot histogram of release date feat
data['release_date'].hist()
plt.title('Game Releases')
plt.ylabel('Number of Games')
plt.xlabel('Year')
plt.show()
```



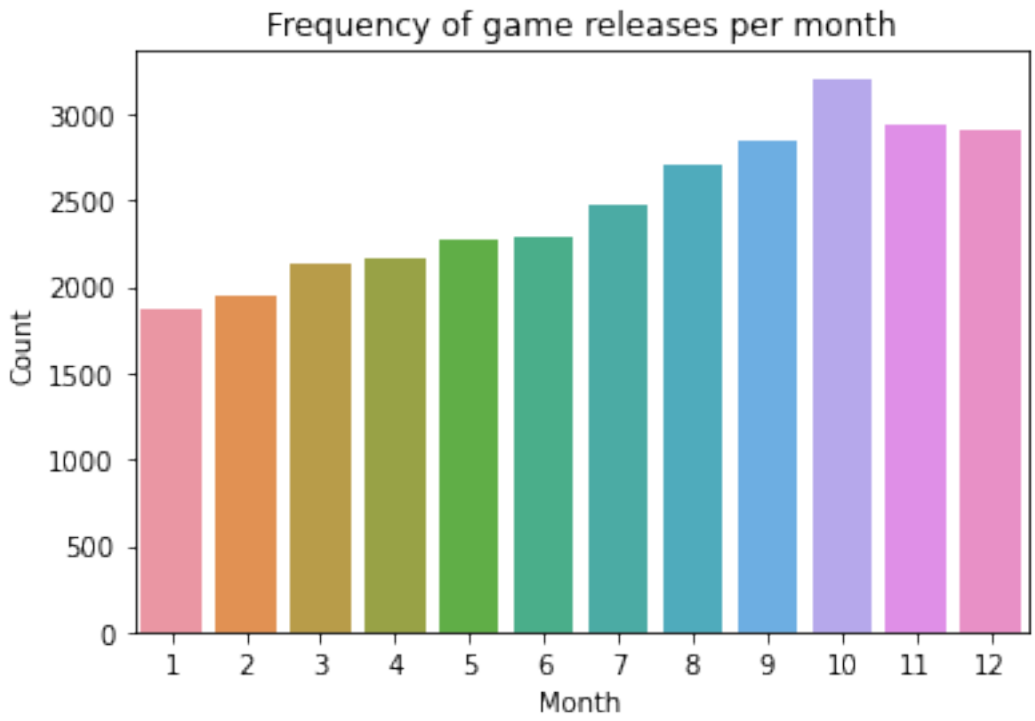
[ ]:
```
# Focus on post 2010
recentgames = data[data['release_date'].dt.year > 2010]

recentgames['release_date'].hist()
plt.title('Game Releases post 2010')
plt.ylabel('Number of Games')
```

```
plt.xlabel('Year')
plt.show()
```

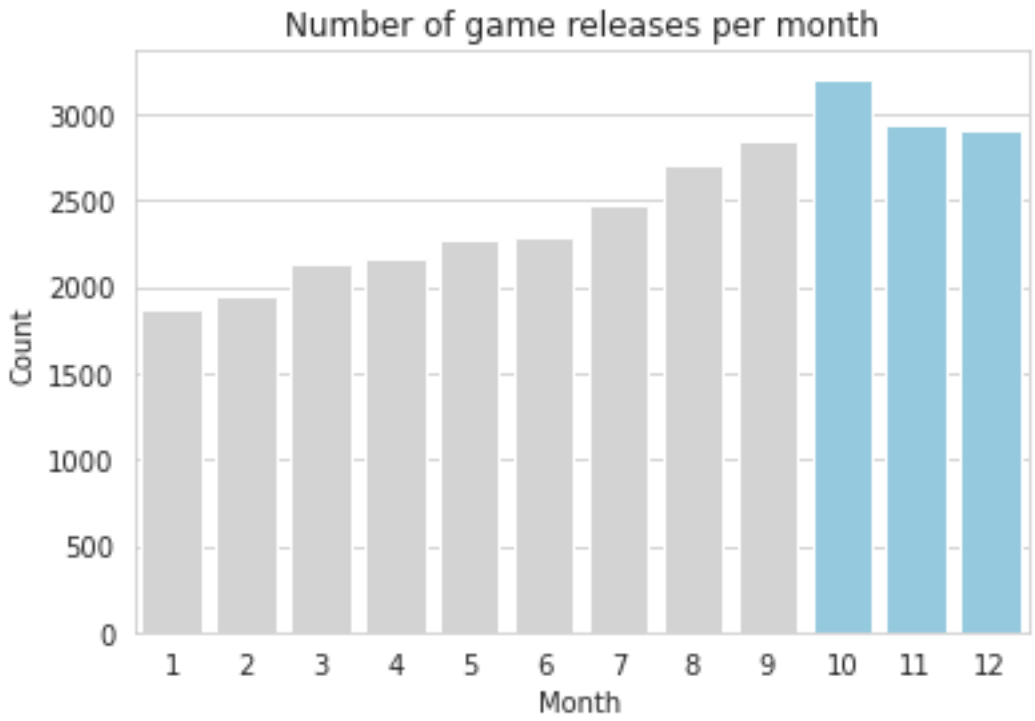


Let's see which months are most popular for new releases.

[ ]:
```
# Create month feature
data['release_month'] = data['release_date'].dt.month

# Plot countplot using Seaborn
sns.countplot(x = data['release_month'], data = data)
plt.title('Frequency of game releases per month')
plt.xlabel('Month')
plt.ylabel('Count')
plt.show()
```



[ ]:
```
# Countplot of sale month

# define palette to highlight best months to buy house
custompalette = {release_month: "skyblue" if (release_month == 10 or␣
 ↪release_month == 11 or release_month == 12 ) else "lightgrey" \
                 for release_month in data['release_month'].unique()}

with sns.axes_style("whitegrid"):
    sns.countplot(x = data['release_month'], palette = custompalette, data =␣
 ↪data)
plt.title('Number of game releases per month')
plt.xlabel('Month')
plt.ylabel('Count')
plt.savefig('Images/month.pdf', bbox_inches = "tight")
```

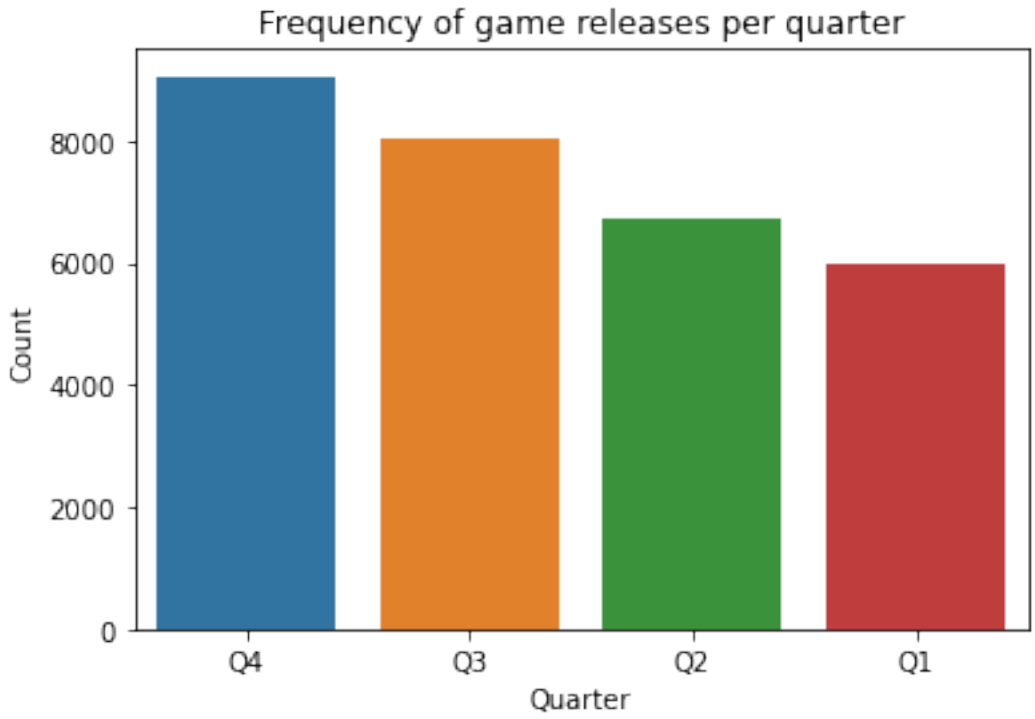


We see that October, November and December have the highest number of game releases. Let's look at quarters now.

[ ]:
```
# Define function to determine quarter
def quarter(month):
    ''' Returns quarter in which month falls'''
    if 1 <= month <= 3:
        quarter = 'Q1'
    elif 4 <= month <= 6:
        quarter = 'Q2'
    elif 7 <= month <= 9:
        quarter = 'Q3'
    else:
        quarter = 'Q4'
    return quarter
```

[ ]:
```
# Create quarter feature
data['release_quarter'] = data['release_month'].apply(quarter)

# Plot countplot using Seaborn
sns.countplot(x = data['release_quarter'], data = data,
              order = data['release_quarter'].value_counts().index)
plt.title('Frequency of game releases per quarter')
plt.xlabel('Quarter')
```

```
plt.ylabel('Count')
plt.show()
```



Recommendation:

Q4 and in particular the month of October sees the most new games released. We would recommend ensuring advertisement deals are priced at a premium during this period.

Finally, let's look at release date for the user-item data.

[ ]:
```
# Create copy to work with
releasedatedata = mergeddata.copy()

# Select entries where release date is not null
releasedatedata = releasedatedata[releasedatedata['release_date'].notnull()]

# Replace strings which are not of the format xxxx-xx-xx with None
releasedatedata['release_date'] = releasedatedata['release_date'].map(lambda x :
 ↪ x if x[-3] == '-'else None)

# Select entries where release date is not null
releasedatedata = releasedatedata[releasedatedata['release_date'].notnull()]

# Convert to DateTime
```

```
releasedatedata['release_date'] = pd.
 ↪to_datetime(releasedatedata['release_date'])

# Check
releasedatedata['release_date'].describe()
```

[ ]:
```
count                  814101
unique                   2598
top       2012-08-21 00:00:00
freq                     7086
first     1983-06-19 00:00:00
last      2018-12-01 00:00:00
Name: release_date, dtype: object
```

Of course, we now have plenty of duplicate entries. However we note that the games span 1983 to 2018.

1.2.3 Game library size

[ ]:
```
# View head
numgames.head()
```

[ ]:

	user_id	items_count
0	76561197970982479	277
1	js41637	888
2	evcentric	137
3	Riot-Punch	328
4	doctr	541

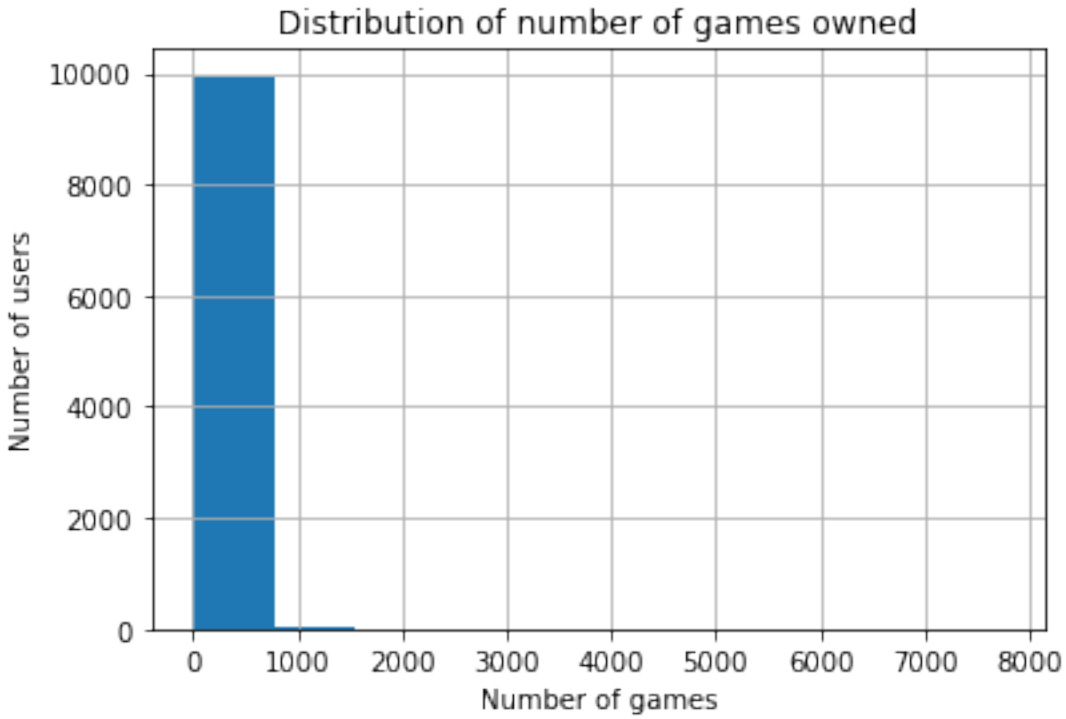
[ ]:
```
# Get summary statistics
numgames['items_count'].describe()
```

[ ]:
```
count    10000.000000
mean        99.498600
std        194.502976
min          0.000000
25%         26.000000
50%         64.000000
75%        121.000000
max       7762.000000
Name: items_count, dtype: float64
```

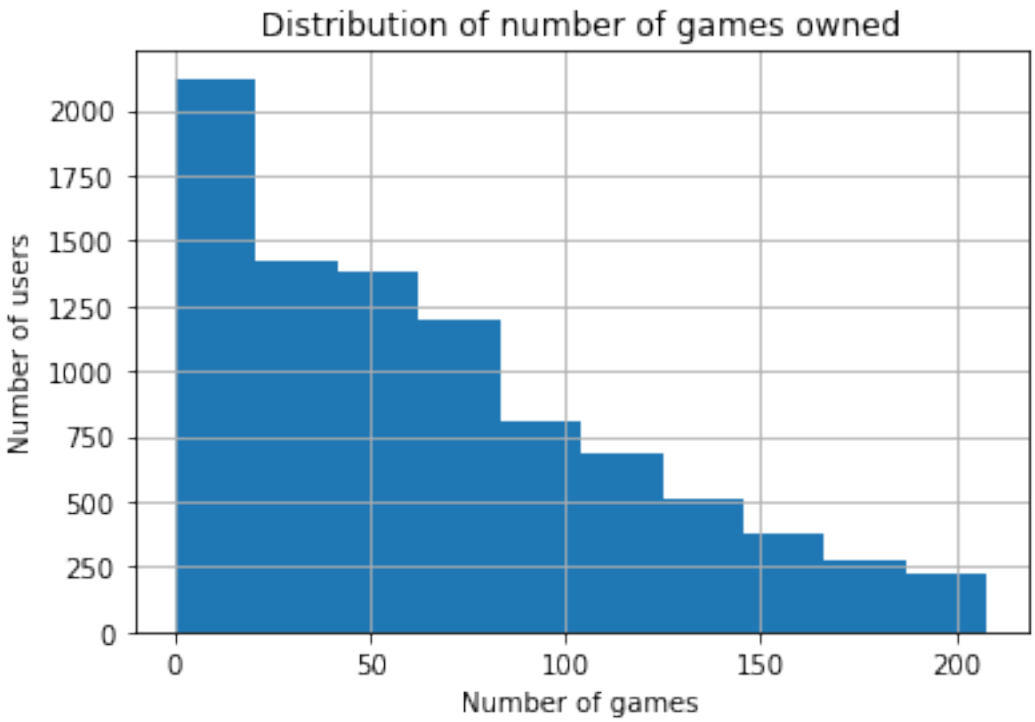
We have data for 88310 unique steam users. We note that the minimum number of games owned is 0 whereas the maximum is 7762. The average number of games owned is 58.

[ ]:
```
# Plot distribution of `items_count`
numgames['items_count'].hist()
plt.title('Distribution of number of games owned')
```

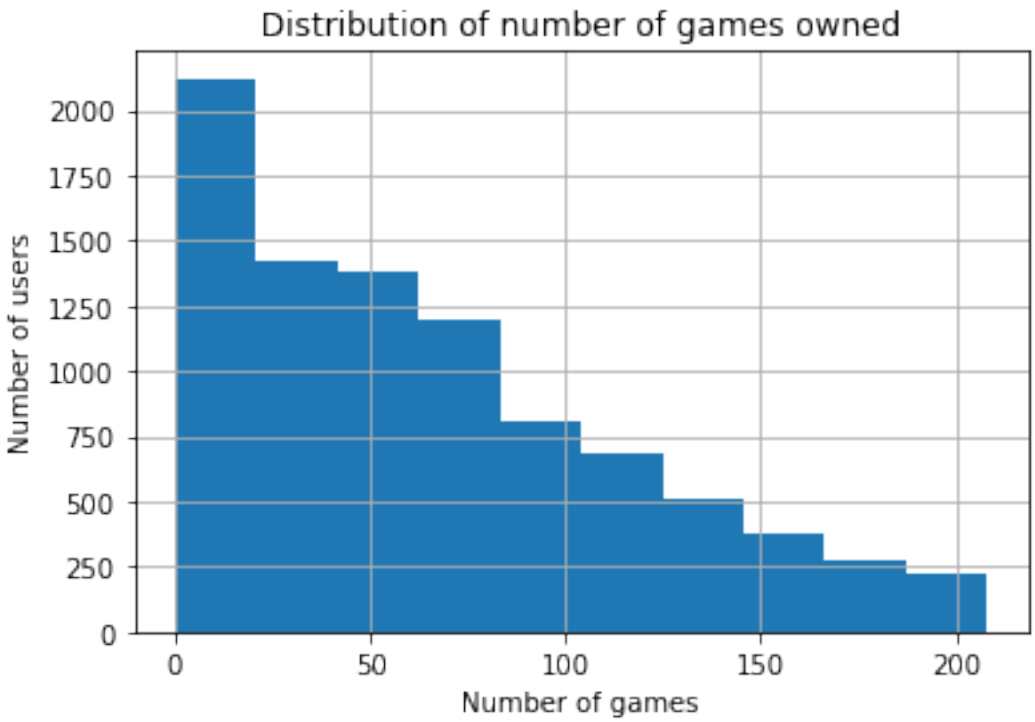
```
plt.xlabel('Number of games')
plt.ylabel('Number of users')
plt.show()
```



[ ]:
```
# Plot distribution of items_count within 90% centile
numgames[numgames['items_count'] < numgames['items_count'].quantile(0.90)].
 ↪hist()
plt.title('Distribution of number of games owned')
plt.xlabel('Number of games')
plt.ylabel('Number of users')
plt.show()
```



[ ]:
```
# Plot distribution of items_count within 90% centile
numgames[numgames['items_count'] < numgames['items_count'].quantile(0.90)].
 ↪hist()
plt.title('Distribution of number of games owned')
plt.xlabel('Number of games')
plt.ylabel('Number of users')
plt.savefig('Images/numgames.pdf', bbox_inches = "tight")
plt.show()
```



Recommendation:

Focus campaign on users who have below the average number of games of 58. These users are more likely to find games they do not own which appeal.

1.2.4 Game Price

[ ]:
```
# Create a copy to work with
gamesprice = gamesdata.copy()

# Get statistics and type
gamesprice['price'].describe()
```

[ ]:
```
count     30758
unique      162
top        4.99
freq       4278
Name: price, dtype: object
```

We see that the values are of type `object`.

From viewing the head above, we noticed the presence of the string `Free To Play`. Let us replace that value with 0.

We will also iterate and replace all strings we find with 0.

[ ]:
```
gamesprice = gamesprice.replace(to_replace = 'Free To Play', value = 0)
gamesprice = gamesprice.replace(to_replace = 'Free to Play', value = 0)
gamesprice = gamesprice.replace(to_replace = 'Free', value = 0)
gamesprice = gamesprice.replace(to_replace = 'Free Demo', value = 0)
gamesprice = gamesprice.replace(to_replace = 'Play for Free!', value = 0)
gamesprice = gamesprice.replace(to_replace = 'Install Now', value = 0)
gamesprice = gamesprice.replace(to_replace = 'Play WARMACHINE: Tactics Demo',␣
 ↪value = 0)
gamesprice = gamesprice.replace(to_replace = 'Free Mod', value = 0)
gamesprice = gamesprice.replace(to_replace = 'Install Theme', value = 0)
gamesprice = gamesprice.replace(to_replace = 'Third-party', value = 0)
gamesprice = gamesprice.replace(to_replace = 'Play Now', value = 0)
gamesprice = gamesprice.replace(to_replace = 'Free HITMAN  Holiday Pack', value␣
 ↪= 0)
gamesprice = gamesprice.replace(to_replace = 'Play the Demo', value = 0)
gamesprice = gamesprice.replace(to_replace = 'Starting at $499.00', value = 0)
gamesprice = gamesprice.replace(to_replace = 'Starting at $449.00', value = 0)
gamesprice = gamesprice.replace(to_replace = 'Free to Try', value = 0)
gamesprice = gamesprice.replace(to_replace = 'Free Movie', value = 0)
gamesprice = gamesprice.replace(to_replace = 'Free to Use', value = 0)
```

[ ]:
```
gamesprice.price.unique()
```

[ ]:
```
array(['4.99', 0, '0.99', '2.99', '3.99', '9.99', '18.99', '29.99', nan,
       '10.99', '1.5899999999999999', '14.99', '1.99', '59.99', '8.99',
       '6.99', '7.99', '39.99', '19.99', '7.49', '12.99', '5.99', '2.49',
       '15.99', '1.25', '24.99', '17.99', '61.99', '3.49', '11.99',
       '13.99', '34.99', '74.76', '1.49', '32.99', '99.99', '14.95',
       '69.99', '16.99', '79.99', '49.99', '5.0', '44.99', '13.98',
       '29.96', '119.99', '109.99', '149.99', '771.71', '21.99', '89.99',
       '0.98', '139.92', '4.29', '64.99', '54.99', '74.99', '0.89', '0.5',
       '299.99', '1.29', '3.0', '15.0', '5.49', '23.99', '49.0', '20.99',
       '10.93', '1.3900000000000001', '36.99', '4.49', '2.0', '4.0',
       '9.0', '234.99', '1.9500000000000002', '1.5', '199.0', '189.0',
       '6.66', '27.99', '10.49', '129.99', '179.0', '26.99', '399.99',
       '31.99', '399.0', '20.0', '40.0', '3.33', '199.99', '22.99',
       '320.0', '38.85', '71.7', '59.95', '995.0', '27.49', '3.39', '6.0',
       '19.95', '499.99', '16.06', '4.68', '131.4', '44.98', '202.76',
       '1.0', '2.3', '0.9500000000000001', '172.24', '249.99',
       '2.9699999999999998', '10.96', '10.0', '30.0', '2.66', '6.48',
       '19.29', '11.15', '18.9', '2.89', '99.0', '87.94', '599.0', '8.98',
       '9.69', '0.49', '9.98', '9.95', '7.0', '12.89', '6.49', '1.87',
       '42.99', '41.99', '289.99', '23.96', '5.65', '12.0', '13.37',
       '189.96', '124.99', '19.98', '160.91'], dtype=object)
```

[ ]:
```
# Convert to float
gamesprice['price'] = gamesprice['price'].astype(float)
```

[ ]:
```
# Get summary statistics
gamesprice['price'].describe()
```

[ ]:
```
count    30758.000000
mean         8.866855
std         15.903457
min          0.000000
25%          2.990000
50%          4.990000
75%          9.990000
max        995.000000
Name: price, dtype: float64
```

We see that 75% of games are under $10! Looks like the majority of games are cheap.

[ ]:
```
belowcentile = gamesprice[gamesprice['price'] < gamesprice['price'].quantile(0.
↪99)]
```

[ ]:
```
belowcentile['price'].describe()
```

[ ]:
```
count    30440.000000
mean         7.879879
std          8.100161
min          0.000000
25%          2.990000
50%          4.990000
75%          9.990000
max         49.990000
Name: price, dtype: float64
```

[ ]:
```
belowcentile['price'].hist()
plt.xlabel('Price in USD')
plt.title('Game Price Distribution')
plt.savefig('Images/price.pdf', bbox_inches = "tight")
plt.show()
```

Recommendation: Focus on volume of sales as the 75% of games are below $10. Highlights the importance of bundles for higher single transactions and where the user may not be interested in all games but still think it worthwhile.

1.2.5 Game genre

[ ]:
```
gamesdata.head()
```

[ ]:
```
          publisher                                              genres  \
0         Kotoshiro  ['Action', 'Casual', 'Indie', 'Simulation', 'S…
1  Making Fun, Inc.       ['Free to Play', 'Indie', 'RPG', 'Strategy']
2     Poolians.com  ['Casual', 'Free to Play', 'Indie', 'Simulatio…
3                                 ['Action', 'Adventure', 'Casual']
4               NaN                                                NaN

                  app_name                    title  \
0     Lost Summoner Kitty      Lost Summoner Kitty
1                Ironbound                Ironbound
2  Real Pool 3D - Poolians  Real Pool 3D - Poolians
3                     2222                     2222
4           Log Challenge                      NaN

                                                 url release_date  \
0  http://store.steampowered.com/app/761140/Lost_…   2018-01-04
```

```
1  http://store.steampowered.com/app/643980/Ironb…   2018-01-04
2  http://store.steampowered.com/app/670290/Real_…   2017-07-24
3     http://store.steampowered.com/app/767400/2222/   2017-12-07
4  http://store.steampowered.com/app/773570/Log_C…          NaN

                                                tags  discount_price  \
0  ['Strategy', 'Action', 'Indie', 'Casual', 'Sim…            4.49
1  ['Free to Play', 'Strategy', 'Indie', 'RPG', '…             NaN
2  ['Free to Play', 'Simulation', 'Sports', 'Casu…             NaN
3                  ['Action', 'Adventure', 'Casual']            0.83
4           ['Action', 'Indie', 'Casual', 'Sports']            1.79

                                          reviews_url  \
0  http://steamcommunity.com/app/761140/reviews/?…
1  http://steamcommunity.com/app/643980/reviews/?…
2  http://steamcommunity.com/app/670290/reviews/?…
3  http://steamcommunity.com/app/767400/reviews/?…
4  http://steamcommunity.com/app/773570/reviews/?…

                                             specs         price  \
0                                ['Single-player']          4.99
1  ['Single-player', 'Multi-player', 'Online Mult…  Free To Play
2  ['Single-player', 'Multi-player', 'Online Mult…  Free to Play
3                                ['Single-player']          0.99
4  ['Single-player', 'Full controller support', '…          2.99

   early_access        id         developer        sentiment  metascore
0         False  761140.0         Kotoshiro              NaN        NaN
1         False  643980.0  Secret Level SRL  Mostly Positive        NaN
2         False  670290.0      Poolians.com  Mostly Positive        NaN
3         False  767400.0                               NaN        NaN
4         False  773570.0               NaN              NaN        NaN
```

[ ]:
```
# Create copy
gamegenres = gamesdata.copy()

# Drop NaN
gamegenres = gamegenres[gamegenres['genres'].notnull()]

# Get unique lists
genres = list(gamegenres['genres'].unique())

# View first 5
genres[:5]
```

[ ]:
```
["['Action', 'Casual', 'Indie', 'Simulation', 'Strategy']",
 "['Free to Play', 'Indie', 'RPG', 'Strategy']",
```

```
 "['Casual', 'Free to Play', 'Indie', 'Simulation', 'Sports']",
 "['Action', 'Adventure', 'Casual']",
 "['Action', 'Adventure', 'Simulation']"]
```

[ ]:
```
# Combine all strings
allgenres = ','.join(genres)

# Preview first 100 characters
allgenres[:100]
```

[ ]:
```
"['Action', 'Casual', 'Indie', 'Simulation', 'Strategy'],['Free to Play',
'Indie', 'RPG', 'Strategy']"
```

[ ]:
```
# Replace chars
allgenres = allgenres.replace("[","").replace("]", "").replace("'", "").
 ↪replace(" ","")

# Check
allgenres[:100]
```

[ ]:
```
'Action,Casual,Indie,Simulation,Strategy,FreetoPlay,Indie,RPG,Strategy,Casual,Fr
eetoPlay,Indie,Simula'
```

[ ]:
```
# Split
splitgenres = allgenres.split(',')
splitgenres[:5]
```

[ ]:
```
['Action', 'Casual', 'Indie', 'Simulation', 'Strategy']
```

[ ]:
```
# Use set to obtain unique values
uniquegenres = set(splitgenres)
uniquegenres
```

[ ]:
```
{'Accounting',
 'Action',
 'Adventure',
 'Animation&amp;amp;Modeling',
 'AudioProduction',
 'Casual',
 'Design&amp;amp;Illustration',
 'EarlyAccess',
 'Education',
 'FreetoPlay',
 'Indie',
 'MassivelyMultiplayer',
 'PhotoEditing',
 'RPG',
 'Racing',
```

```
 'Simulation',
 'SoftwareTraining',
 'Sports',
 'Strategy',
 'Utilities',
 'VideoProduction',
 'WebPublishing'}
```

[ ]:
```
# Create columns with genres
for genre in uniquegenres:
    gamegenres[genre] = 0

# Split genres in genres column
gamegenres['genres'] = gamegenres['genres'].map(lambda x : x.replace("[","").
 ↪replace("]", "").replace("'", "").replace(" ","").split(','))

# Map to columns - set to 1 if genre applies
for index, genres in enumerate(gamegenres['genres']):
    for genre in genres:
        gamegenres.loc[index,genre] = 1

# Visuale the new columns
gamegenres.head(2)
```

[ ]:
```
          publisher                                          genres  \
0         Kotoshiro  [Action, Casual, Indie, Simulation, Strategy]
1  Making Fun, Inc.             [FreetoPlay, Indie, RPG, Strategy]

              app_name                title  \
0  Lost Summoner Kitty  Lost Summoner Kitty
1            Ironbound            Ironbound

                                                 url release_date  \
0  http://store.steampowered.com/app/761140/Lost_…   2018-01-04
1  http://store.steampowered.com/app/643980/Ironb…   2018-01-04

                                               tags  discount_price  \
0  ['Strategy', 'Action', 'Indie', 'Casual', 'Sim…            4.49
1  ['Free to Play', 'Strategy', 'Indie', 'RPG', '…             NaN

                                          reviews_url  \
0  http://steamcommunity.com/app/761140/reviews/?…
1  http://steamcommunity.com/app/643980/reviews/?…

                                             specs  …  \
0                                ['Single-player']  …
1  ['Single-player', 'Multi-player', 'Online Mult…  …
```

```
  Animation&amp;amp;Modeling Racing  Casual Education EarlyAccess  Utilities  \
0                        0.0    0.0     1.0       0.0         0.0        0.0
1                        0.0    0.0     0.0       0.0         0.0        0.0

   WebPublishing  PhotoEditing  Design&amp;amp;Illustration  Indie
0            0.0           0.0                        0.0    1.0
1            0.0           0.0                        0.0    1.0

[2 rows x 38 columns]
```

[ ]:
```
gamegenres.columns
```

[ ]:
```
Index(['publisher', 'genres', 'app_name', 'title', 'url', 'release_date',
       'tags', 'discount_price', 'reviews_url', 'specs', 'price',
       'early_access', 'id', 'developer', 'sentiment', 'metascore',
       'Adventure', 'Sports', 'RPG', 'FreetoPlay', 'SoftwareTraining',
       'VideoProduction', 'AudioProduction', 'Simulation', 'Action',
       'Strategy', 'MassivelyMultiplayer', 'Accounting',
       'Animation&amp;amp;Modeling', 'Racing', 'Casual', 'Education',
       'EarlyAccess', 'Utilities', 'WebPublishing', 'PhotoEditing',
       'Design&amp;amp;Illustration', 'Indie'],
      dtype='object')
```

[ ]:
```
# Start with empty dictionary
genredict = {}

# Get genre columns
genrecols = gamegenres.loc[:, 'Adventure':'Indie'].columns

# Go through each column and sum it
for col in genrecols:
    genredict[col] = gamegenres[col].sum()

# sort dictionary based on counts, ascending order so reverse = True
sortedgenresdict = {keys: values for keys, values in \
                        sorted(genredict.items(), key = lambda item: item[1],␣
 ↪reverse = True)}
```

[ ]:
```
# View dictionary
sortedgenresdict
```

[ ]:
```
{'Indie': 15858.0,
 'Action': 11321.0,
 'Casual': 8282.0,
 'Adventure': 8243.0,
 'Strategy': 6957.0,
```

```
 'Simulation': 6699.0,
 'RPG': 5479.0,
 'FreetoPlay': 2031.0,
 'EarlyAccess': 1462.0,
 'Sports': 1257.0,
 'MassivelyMultiplayer': 1108.0,
 'Racing': 1083.0,
 'Design&amp;amp;Illustration': 460.0,
 'Utilities': 340.0,
 'WebPublishing': 268.0,
 'Animation&amp;amp;Modeling': 183.0,
 'Education': 125.0,
 'VideoProduction': 116.0,
 'SoftwareTraining': 105.0,
 'AudioProduction': 93.0,
 'PhotoEditing': 77.0,
 'Accounting': 7.0}
```

We see that Indie is the most popular genre, followed by Action. On the other end of the spectrum, there are few entries relating to Photo Editing and only 7 for Accounting. This makes sense as Steam is a gaming platform, and so photo editing or accounting software doesn't really belong.

1.2.6 Game tags

[ ]:
```
# Create copy
gametags = gamesdata.copy()

# Drop NaN
gametags = gamegenres[gamegenres['tags'].notnull()]

# Get unique lists
tags = list(gametags['tags'].unique())

# View first 5
tags[:5]
```

[ ]:
```
["['Strategy', 'Action', 'Indie', 'Casual', 'Simulation']",
 "['Free to Play', 'Strategy', 'Indie', 'RPG', 'Card Game', 'Trading Card Game',
'Turn-Based', 'Fantasy', 'Tactical', 'Dark Fantasy', 'Board Game', 'PvP', '2D',
'Competitive', 'Replay Value', 'Character Customization', 'Female Protagonist',
'Difficult', 'Design & Illustration']",
 "['Free to Play', 'Simulation', 'Sports', 'Casual', 'Indie', 'Multiplayer']",
 "['Action', 'Adventure', 'Casual']",
 "['Action', 'Adventure', 'Simulation', 'FPS', 'Shooter', 'Third-Person
Shooter', 'Sniper', 'Third Person']"]
```

[ ]:
```
# Combine all strings
alltags = ','.join(tags)

# Preview first 100 characters
alltags[:100]
```

[ ]:
```
"['Strategy', 'Action', 'Indie', 'Casual', 'Simulation'],['Free to Play',
'Strategy', 'Indie', 'RPG',"
```

[ ]:
```
# Replace chars
alltags = alltags.replace("[","").replace("]", "").replace("'", "")

# Check
alltags[:100]
```

[ ]:
```
' Strategy, Action, Indie, Casual, Simulation, Free to Play, Strategy, Indie,
RPG, Card Game, Trading'
```

[ ]:
```
# Split
splittags = alltags[1:].split(',')
splittags[:5]
```

[ ]:
```
['Strategy', ' Action', ' Indie', ' Casual', ' Simulation']
```

[ ]:
```
# Use set to obtain unique values
uniquetags = set(splittags)
len(uniquetags)
```

[ ]:
```
337
```

1.2.7 Top publishers

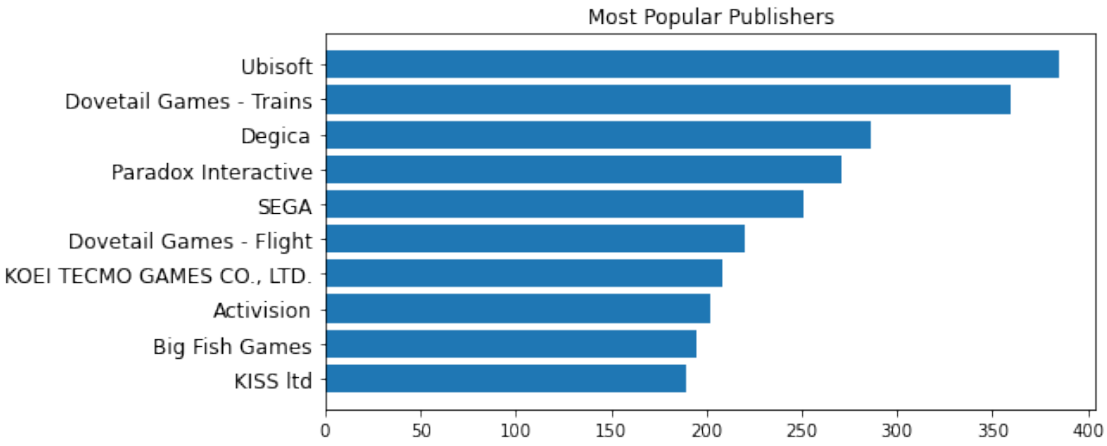
[ ]:
```
# Select entries where publisher is non-null
data = gamesdata[gamesdata['publisher'].notnull()]
```

[ ]:
```
# Create dictionary
game_publishers = {}
for publisher in list(data['publisher']):
    if not publisher in game_publishers:
        game_publishers[publisher] = 1
    else:
        game_publishers[publisher] += 1
```

[ ]:
```
# Get top 10 publishers
top10_publishers = dict(Counter(game_publishers).most_common(10))
top10_publishers
```

[ ]:
```
{'Ubisoft': 385,
 'Dovetail Games - Trains': 360,
 'Degica': 286,
 'Paradox Interactive': 271,
 'SEGA': 251,
 'Dovetail Games - Flight': 220,
 'KOEI TECMO GAMES CO., LTD.': 208,
 'Activision': 202,
 'Big Fish Games': 195,
 'KISS ltd': 189}
```

[ ]:
```
# Prepare for bar chart plot
top10_publishers = dict(sorted(Counter(game_publishers).most_common(10),␣
 ↪key=lambda x:x[1]))

# Plots most popular publishers
fig = plt.figure(figsize = (8,4))
plt.barh(range(len(top10_publishers)), list(top10_publishers.values()),␣
 ↪align='center')
plt.yticks(range(len(top10_publishers)), list(top10_publishers.keys()),␣
 ↪fontsize=12)
plt.title("Most Popular Publishers", fontsize=12, fontweight= 22)
plt.show()
```



[ ]:

Part-4-Model-Based

December 21, 2021

[ ]:
```
import pandas as pd
import numpy as np
pd.set_option('max_rows', 15)
pd.plotting.register_matplotlib_converters()
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
from sklearn.metrics import mean_absolute_error
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeRegressor
```

[ ]:
```
# Load merged data
mergeddata = pd.read_csv('mergeddata.csv', index_col = 0)
mergeddata.head()
```

```
C:\Users\pmogh\anaconda3\lib\site-
packages\IPython\core\interactiveshell.py:3185: DtypeWarning: Columns (14) have
mixed types.Specify dtype option on import or set low_memory=False.
  has_raised = await self.run_ast_nodes(code_ast.body, cell_name,
```

[ ]:

	uid	id	owned	publisher	genres	app_name	title	url	release_date	tags	discount_price
0	0	10	1.0	Valve	['Action']	Counter-Strike	Counter-Strike	http://store.steampowered.com/app/10/CounterSt…	2000-11-01	['Action', 'FPS', 'Multiplayer', 'Shooter', 'C…	NaN
1	1	10	1.0	Valve	['Action']	Counter-Strike	Counter-Strike	http://store.steampowered.com/app/10/CounterSt…	2000-11-01	['Action', 'FPS', 'Multiplayer', 'Shooter', 'C…	NaN
2	3	10	1.0	Valve	['Action']	Counter-Strike	Counter-Strike	http://store.steampowered.com/app/10/CounterSt…	2000-11-01	['Action', 'FPS', 'Multiplayer', 'Shooter', 'C…	NaN
3	4	10	1.0	Valve	['Action']	Counter-Strike	Counter-Strike	http://store.steampowered.com/app/10/CounterSt…	2000-11-01		
4	10	10	1.0	Valve	['Action']	Counter-Strike	Counter-Strike	http://store.steampowered.com/app/10/CounterSt…	2000-11-01		

```
3  ['Action', 'FPS', 'Multiplayer', 'Shooter', 'C…             NaN
4  ['Action', 'FPS', 'Multiplayer', 'Shooter', 'C…             NaN

                                        reviews_url  \
0  http://steamcommunity.com/app/10/reviews/?brow…
1  http://steamcommunity.com/app/10/reviews/?brow…
2  http://steamcommunity.com/app/10/reviews/?brow…
3  http://steamcommunity.com/app/10/reviews/?brow…
4  http://steamcommunity.com/app/10/reviews/?brow…

                                          specs price  early_access developer  \
0  ['Multi-player', 'Valve Anti-Cheat enabled']  9.99         False     Valve
1  ['Multi-player', 'Valve Anti-Cheat enabled']  9.99         False     Valve
2  ['Multi-player', 'Valve Anti-Cheat enabled']  9.99         False     Valve
3  ['Multi-player', 'Valve Anti-Cheat enabled']  9.99         False     Valve
4  ['Multi-player', 'Valve Anti-Cheat enabled']  9.99         False     Valve

                 sentiment  metascore
0  Overwhelmingly Positive       88.0
1  Overwhelmingly Positive       88.0
2  Overwhelmingly Positive       88.0
3  Overwhelmingly Positive       88.0
4  Overwhelmingly Positive       88.0
```

[ ]:
```
mergeddata.shape
```

[ ]:
```
(819988, 18)
```

[ ]:
```
mergeddata.describe()
```

[ ]:
```
                uid             id     owned  discount_price      metascore
count  819988.000000  819988.000000  819988.0     1052.000000  485063.000000
mean     5090.851825  185177.375136       1.0        0.637405      78.504638
std      2858.808048  135141.482185       0.0        0.698054       9.801869
min         0.000000      10.000000       1.0        0.490000      20.000000
25%      2641.000000   34900.000000       1.0        0.490000      73.000000
50%      5121.000000  218820.000000       1.0        0.490000      80.000000
75%      7546.000000  287290.000000       1.0        0.490000      85.000000
max      9999.000000  530720.000000       1.0        7.490000      96.000000
```

[ ]:
```
mergeddata.describe(include="object")
```

[ ]:
```
       publisher      genres                          app_name  \
count     806910      816219                            819988
unique      3747         569                              8169
top        Valve  ['Action']  Counter-Strike: Global Offensive
freq       58250      145396                              6966
```

```
                                  title  \
count                            819988
unique                             8169
top     Counter-Strike: Global Offensive
freq                               6966

                                                    url release_date  \
count                                            819988       818866
unique                                             8171         2621
top     http://store.steampowered.com/app/730/CounterS…   2012-08-21
freq                                               6966         7086

                                                    tags  \
count                                             819971
unique                                              7125
top     ['FPS', 'Multiplayer', 'Shooter', 'Action', 'T…
freq                                               6966

                                           reviews_url              specs  \
count                                           819988             816228
unique                                            8171               1563
top     http://steamcommunity.com/app/730/reviews/?bro…  ['Single-player']
freq                                              6966              68096

         price developer      sentiment
count   804282    812896         819639
unique     104      5228             18
top       9.99     Valve  Very Positive
freq    136063     49265         410388
```

1 Find the count of Numeric column and Categorical Column

[ ]:
```
print(f"Find the count of Numeric Column : {len(mergeddata.
 ↪select_dtypes(include=np.number).columns)}")

print(f"Numeric Column Name : {mergeddata.select_dtypes(include=np.number).
 ↪columns.tolist}")
```

```
Find the count of Numeric Column : 5
Numeric Column Name : <bound method IndexOpsMixin.tolist of Index(['uid', 'id',
'owned', 'discount_price', 'metascore'], dtype='object')>
```

[ ]:
```
print(f"Find the count of Categorical Column : {len(mergeddata.
 ↪select_dtypes(exclude=np.number).columns)}")
```

```
print(f"Categorical Column Name : {mergeddata.select_dtypes(exclude=np.number).
 ↪columns.tolist}")
```

```
Find the count of Categorical Column : 13
Categorical Column Name : <bound method IndexOpsMixin.tolist of
Index(['publisher', 'genres', 'app_name', 'title', 'url', 'release_date',
       'tags', 'reviews_url', 'specs', 'price', 'early_access', 'developer',
       'sentiment'],
      dtype='object')>
```

[ ]:
```
mergeddata.isnull().sum().sort_values(ascending=False).head(15)
```

[ ]:
```
discount_price    818936
metascore         334925
price              15706
publisher          13078
developer           7092
genres              3769
specs               3760
release_date        1122
sentiment            349
tags                  17
id                     0
reviews_url            0
url                    0
title                  0
early_access           0
dtype: int64
```

2 Find total null values in data and percentage of null values in each columns

[ ]:
```
def find_total_perc_missing (data_set):
    temp_missing_val = (data_set.isnull().sum()).sum()
    total_cel = np.product(data_set.shape)
    perc_missing_data=100 * (temp_missing_val/total_cel)
    return perc_missing_data
```

[ ]:
```
print(find_total_perc_missing(mergeddata))
```

```
8.121758421396953
```

[ ]:
```
def find_missing_value(data_set):
    percent_missing = data_set.isnull().sum() * 100 / len(data_set)
    missing_value_df = pd.DataFrame({'column_name': data_set.
 ↪columns,'percent_missing': percent_missing})
```

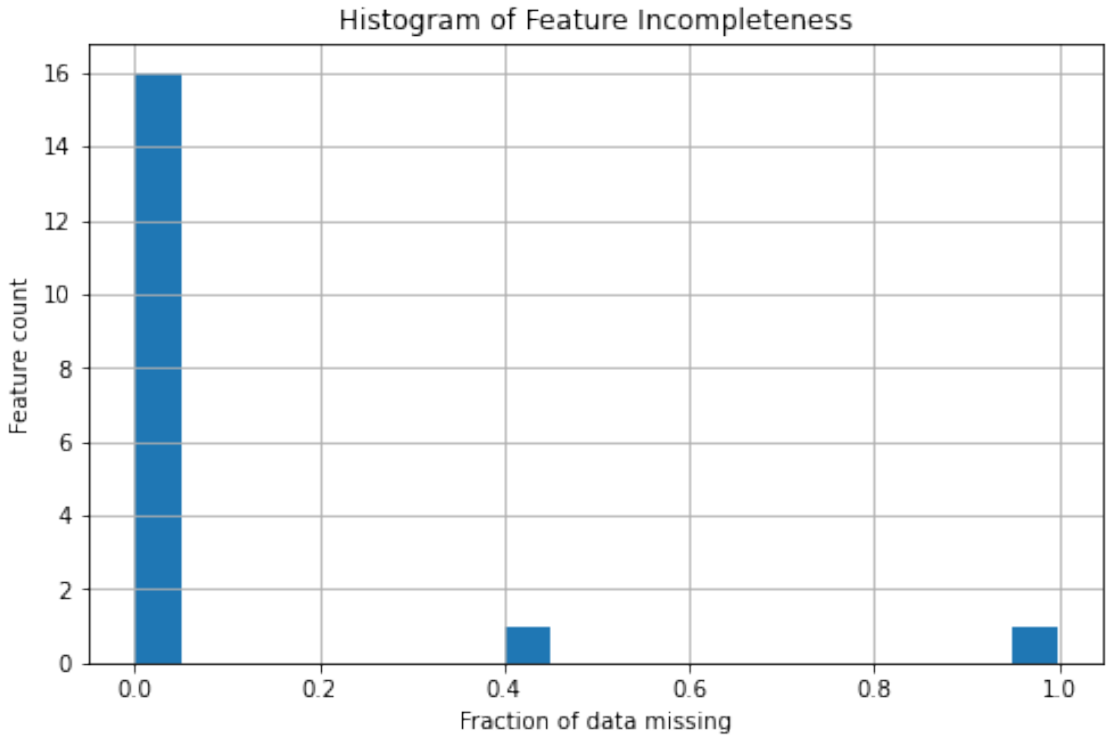
```
    missing_value_df=missing_value_df.sort_values('percent_missing',␣
↪ascending=False)

    return missing_value_df
```

[ ]:
```
find_missing_value(mergeddata).head(15)
```

[ ]:

	column_name	percent_missing
discount_price	discount_price	99.871705
metascore	metascore	40.845110
price	price	1.915394
publisher	publisher	1.594901
developer	developer	0.864891
genres	genres	0.459641
specs	specs	0.458543
release_date	release_date	0.136831
sentiment	sentiment	0.042562
tags	tags	0.002073
id	id	0.000000
reviews_url	reviews_url	0.000000
url	url	0.000000
title	title	0.000000
early_access	early_access	0.000000

[ ]:
```
plt.figure(figsize=(8,5))
(mergeddata.isna().sum() / mergeddata.shape[0]).hist(bins=20)
plt.title('Histogram of Feature Incompleteness')
plt.xlabel('Fraction of data missing')
plt.ylabel('Feature count');
```



[ ]:
```
def remove_columns (data_set,final_val_data):
    list_remove_column=[]
    for index,row in final_val_data.iterrows():
        if row['percent_missing'] > 60:
            if index in mergeddata.columns:
                list_remove_column.append(index)
    return list_remove_column
```

3 Remove the column with more than 50% null values

[ ]:
```
# calling the above function to find the list of above 60
Removed_Column_Name=remove_columns(mergeddata,find_missing_value(mergeddata))
mergeddata=mergeddata.drop(Removed_Column_Name, axis=1)
```

[ ]:
```
find_missing_value(mergeddata).head(15)
```

[ ]:

	column_name	percent_missing
metascore	metascore	40.845110
price	price	1.915394
publisher	publisher	1.594901
developer	developer	0.864891
genres	genres	0.459641

```
specs                specs         0.458543
release_date  release_date         0.136831
sentiment        sentiment         0.042562
tags                  tags         0.002073
url                    url         0.000000
id                      id         0.000000
reviews_url    reviews_url         0.000000
title                title         0.000000
app_name          app_name         0.000000
early_access  early_access         0.000000
```

Now check the shape of the dataset

[ ]:
```
print('Merge Data',mergeddata.isnull().all(axis=0).sum())
```

```
Merge Data 0
```

[ ]:
```
print("Merge Data",len(mergeddata[mergeddata.isnull().sum(axis=1)>14].index))
```

```
Merge Data 0
```

[ ]:
```
type(mergeddata.release_date)
```

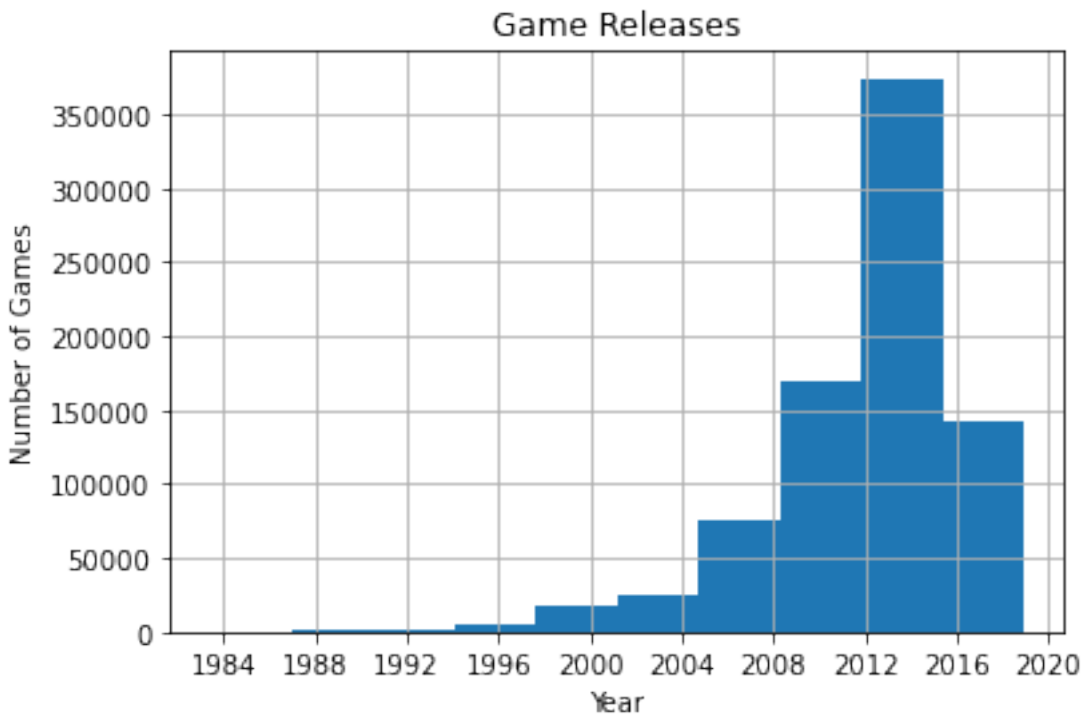
[ ]:
```
pandas.core.series.Series
```

Feature engineer the date into year and months

[ ]:
```
mergeddata['release_date'] = pd.to_datetime(pd.
 ↪to_datetime(mergeddata['release_date'], errors='coerce', format='%Y-%m-%d'))
```

[ ]:
```
mergeddata['release_date_year']=mergeddata['release_date'].dt.year
mergeddata['release_date_month']=mergeddata['release_date'].dt.month
```

[ ]:
```
# Plot histogram of release date feat
mergeddata['release_date'].hist()
plt.title('Game Releases')
plt.ylabel('Number of Games')
plt.xlabel('Year')
plt.show()
```



[ ]:
```
mergeddata.shape
```

[ ]:
```
(819988, 19)
```

[ ]:
```
mergeddata.sentiment.value_counts()
```

[ ]:
```
Very Positive              410388
Overwhelmingly Positive    133583
Mixed                      124518
Mostly Positive            118654
Mostly Negative             13260
                             ...
5 user reviews                682
3 user reviews                646
9 user reviews                608
Overwhelmingly Negative       539
1 user reviews                523
Name: sentiment, Length: 18, dtype: int64
```

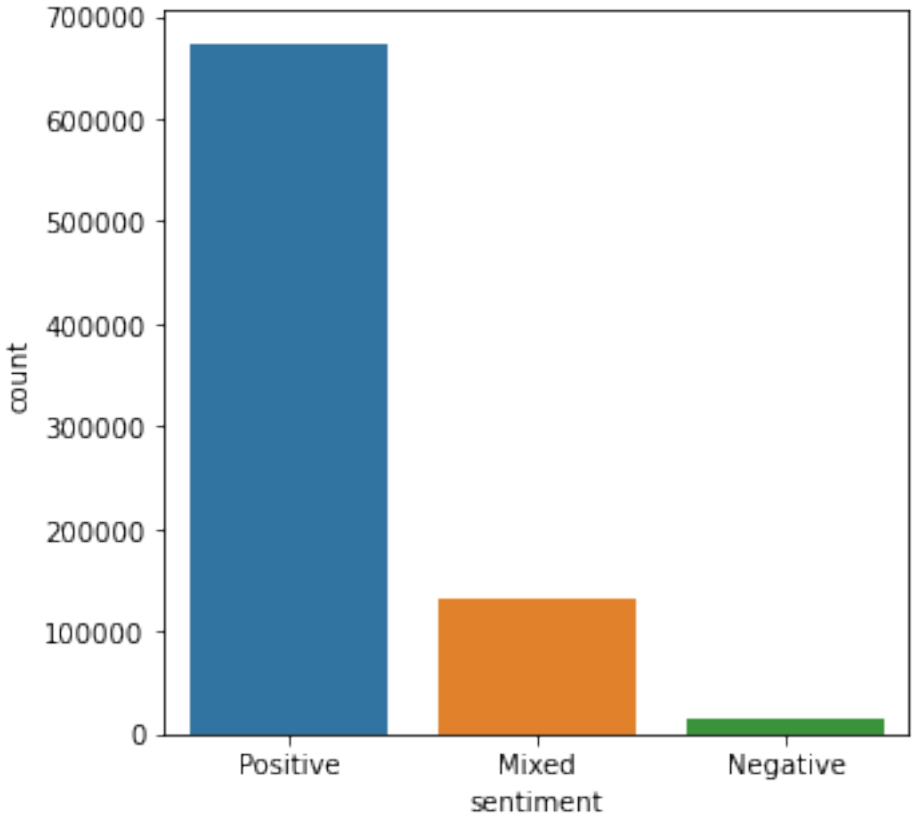
[ ]:
```
mergeddata.sentiment.unique()
```

[ ]:
```
array(['Overwhelmingly Positive', 'Very Positive', 'Mostly Positive',
       'Mixed', 'Mostly Negative', 'Overwhelmingly Negative', 'Positive',
       '2 user reviews', '8 user reviews', '5 user reviews',
```

```
        '7 user reviews', '6 user reviews', 'Very Negative',
        '3 user reviews', 'Negative', '4 user reviews', '1 user reviews',
        '9 user reviews', nan], dtype=object)
```

[ ]:
```
mergeddata['sentiment']=mergeddata['sentiment'].replace(['Overwhelmingly␣
 ↪Positive','Very Positive','Mostly Positive', 'Positive'],'Positive')
mergeddata['sentiment']=mergeddata['sentiment'].replace(['Overwhelmingly␣
 ↪Negative','Very Negative','Mostly Negative', 'Negative'],'Negative')
mergeddata['sentiment']=mergeddata['sentiment'].replace(['1 user reviews','2␣
 ↪user reviews','3 user reviews', '4 user reviews','5 user reviews','6 user␣
 ↪reviews','7 user reviews', '8 user reviews','9 user reviews'],'Mixed')
```

[ ]:
```
plt.figure(figsize=(5,5))
sns.countplot(x='sentiment', data=mergeddata)
plt.show()
```



Filling the Missing Data to remove null values.

[ ]:
```
from sklearn.base import TransformerMixin

class DataFrameImputer(TransformerMixin):
```

```
    def __init__(self):
        """Impute missing values.

        Columns of dtype object are imputed with the most frequent value
        in column.

        Columns of other types are imputed with mean of column.

        """
    def fit(self, X, y=None):
        # Find most common value with value_counts() which returns
        # counts in descending order so that the first element is the most␣
↪frequently-occurring element.
        self.fill = pd.Series([X[c].value_counts().index[0]
            #Use that if type is object otherwise use mean
            if X[c].dtype == np.dtype('O') else X[c].mean() for c in X],
            index=X.columns)

        return self

    def transform(self, X, y=None):
        return X.fillna(self.fill)
```

[ ]:
```
find_missing_value(mergeddata)
```

[ ]:
```
                          column_name  percent_missing
metascore                   metascore        40.845110
price                           price         1.915394
publisher                   publisher         1.594901
developer                   developer         0.864891
release_date_month  release_date_month         0.717937
…                                  …              …
url                               url         0.000000
title                           title         0.000000
app_name                     app_name         0.000000
owned                           owned         0.000000
uid                               uid         0.000000

[19 rows x 2 columns]
```

[ ]:
```
mergeddata = DataFrameImputer().fit_transform(mergeddata)
```

[ ]:
```
find_missing_value(mergeddata)
```

[ ]:
```
                          column_name  percent_missing
uid                               uid              0.0
```

```
reviews_url                reviews_url              0.0
release_date_year    release_date_year              0.0
metascore                  metascore                0.0
sentiment                  sentiment                0.0
…                              …                  …
app_name                    app_name                0.0
genres                        genres                0.0
publisher                  publisher                0.0
owned                          owned                0.0
release_date_month  release_date_month              0.0

[19 rows x 2 columns]
```

[ ]:
```
svd_data=mergeddata.copy(deep=True)
```

[ ]:
```
mergeddata.head()
```

[ ]:
```
   uid  id  owned publisher      genres        app_name           title  \
0    0  10    1.0     Valve  ['Action']  Counter-Strike  Counter-Strike
1    1  10    1.0     Valve  ['Action']  Counter-Strike  Counter-Strike
2    3  10    1.0     Valve  ['Action']  Counter-Strike  Counter-Strike
3    4  10    1.0     Valve  ['Action']  Counter-Strike  Counter-Strike
4   10  10    1.0     Valve  ['Action']  Counter-Strike  Counter-Strike

                                                 url release_date  \
0  http://store.steampowered.com/app/10/CounterSt…   2000-11-01
1  http://store.steampowered.com/app/10/CounterSt…   2000-11-01
2  http://store.steampowered.com/app/10/CounterSt…   2000-11-01
3  http://store.steampowered.com/app/10/CounterSt…   2000-11-01
4  http://store.steampowered.com/app/10/CounterSt…   2000-11-01

                                               tags  \
0  ['Action', 'FPS', 'Multiplayer', 'Shooter', 'C…
1  ['Action', 'FPS', 'Multiplayer', 'Shooter', 'C…
2  ['Action', 'FPS', 'Multiplayer', 'Shooter', 'C…
3  ['Action', 'FPS', 'Multiplayer', 'Shooter', 'C…
4  ['Action', 'FPS', 'Multiplayer', 'Shooter', 'C…

                                        reviews_url  \
0  http://steamcommunity.com/app/10/reviews/?brow…
1  http://steamcommunity.com/app/10/reviews/?brow…
2  http://steamcommunity.com/app/10/reviews/?brow…
3  http://steamcommunity.com/app/10/reviews/?brow…
4  http://steamcommunity.com/app/10/reviews/?brow…

                                      specs price  early_access developer  \
0  ['Multi-player', 'Valve Anti-Cheat enabled']  9.99         False     Valve
```

```
1  ['Multi-player', 'Valve Anti-Cheat enabled']  9.99         False     Valve
2  ['Multi-player', 'Valve Anti-Cheat enabled']  9.99         False     Valve
3  ['Multi-player', 'Valve Anti-Cheat enabled']  9.99         False     Valve
4  ['Multi-player', 'Valve Anti-Cheat enabled']  9.99         False     Valve
```

	sentiment	metascore	release_date_year	release_date_month
0	Positive	88.0	2000.0	11.0
1	Positive	88.0	2000.0	11.0
2	Positive	88.0	2000.0	11.0
3	Positive	88.0	2000.0	11.0
4	Positive	88.0	2000.0	11.0

[ ]:
```
print(mergeddata.select_dtypes(exclude=np.number).columns.tolist)
```

```
<bound method IndexOpsMixin.tolist of Index(['publisher', 'genres', 'app_name',
'title', 'url', 'release_date',
       'tags', 'reviews_url', 'specs', 'price', 'early_access', 'developer',
       'sentiment'],
      dtype='object')>
```

[ ]:
```
import category_encoders as ce

def encode_category_data(dataset_var):
       #We establish the Ordinal encoder which will convert each categorical␣
 ↪label to a number
       # We specify the columns we want to transform, we ask it to handle␣
 ↪missing values if any and also to return a dataframe                    ␣
 ↪instead of an np array
       encode_var = ce.OrdinalEncoder(cols=['publisher', 'genres', 'app_name',␣
 ↪'title', 'url', 'release_date',
       'tags', 'reviews_url', 'specs', 'price', 'early_access', 'developer',
       'sentiment'],handle_missing='return_nan',return_df= True)
       dataset_var=encode_var.fit_transform(dataset_var)
       return dataset_var
```

[ ]:
```
#We now fit the model and transform the data and put it in X which is a␣
 ↪dataframe
mergeddata=encode_category_data(mergeddata)
```

[ ]:
```
mergeddata.sentiment.value_counts()
```

[ ]:
```
1.0    673232
2.0    131330
3.0     15426
Name: sentiment, dtype: int64
```

[ ]:
```
mergeddata.head()
```

[ ]:

	uid	id	owned	publisher	genres	app_name	title	url	release_date	\
0	0	10	1.0	1.0	1.0	1.0	1.0	1.0	1.0	
1	1	10	1.0	1.0	1.0	1.0	1.0	1.0	1.0	
2	3	10	1.0	1.0	1.0	1.0	1.0	1.0	1.0	
3	4	10	1.0	1.0	1.0	1.0	1.0	1.0	1.0	
4	10	10	1.0	1.0	1.0	1.0	1.0	1.0	1.0	

	tags	reviews_url	specs	price	early_access	developer	sentiment	\
0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	
1	1.0	1.0	1.0	1.0	1.0	1.0	1.0	
2	1.0	1.0	1.0	1.0	1.0	1.0	1.0	
3	1.0	1.0	1.0	1.0	1.0	1.0	1.0	
4	1.0	1.0	1.0	1.0	1.0	1.0	1.0	

	metascore	release_date_year	release_date_month
0	88.0	2000.0	11.0
1	88.0	2000.0	11.0
2	88.0	2000.0	11.0
3	88.0	2000.0	11.0
4	88.0	2000.0	11.0

Due to hardware Limitation,I am only taking 2Lakh Data for Model

[ ]:
```
temp = 200000
mergeddata=mergeddata[:temp]
```

[ ]:
```
mergeddata.sentiment.value_counts()
```

[ ]:
```
1.0    191818
2.0      5693
3.0      2489
Name: sentiment, dtype: int64
```

Choose the Y and x for the Split

[ ]:
```
X = mergeddata.iloc[:,mergeddata.columns!='sentiment']
y = mergeddata.iloc[:,mergeddata.columns=='sentiment']
print(y.shape)
print(X.shape)
del mergeddata
```

```
(200000, 1)
(200000, 18)
```

Apply the K-Fold to split the Test and Train Equally

[ ]:
```
from sklearn.model_selection import StratifiedKFold
variables = StratifiedKFold(n_splits=4)
```

```
for train,test in variables.split(X,y):
    x_train,x_test=X.iloc[train],X.iloc[test]
    y_train,y_test=y.iloc[train],y.iloc[test]
```

[ ]:
```
from sklearn.metrics import accuracy_score, balanced_accuracy_score, f1_score
from sklearn.metrics import␣
↪confusion_matrix,plot_confusion_matrix,classification_report
from sklearn.metrics import accuracy_score
from imblearn.over_sampling import SMOTE
from sklearn.preprocessing import Normalizer
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import GridSearchCV,RandomizedSearchCV

from sklearn.pipeline import Pipeline
from sklearn.decomposition import PCA
from sklearn import tree
from sklearn import svm
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from surprise import SVD
from surprise import Dataset
from surprise.model_selection import train_test_split,cross_validate
from surprise import Reader ,accuracy
```

[ ]:
```
def␣
↪check_standalone_prediction(smote_flag,norm_flag,scale_flag,pca_flag,x_train,y_train,x_test
↪
    if smote_flag == True:
        sm=SMOTE(random_state=42)
        x_train, y_train = sm.fit_resample(x_train, y_train)
    if norm_flag == True:
        normalizer = Normalizer(norm='l2')
        x_train = normalizer.fit_transform(x_train)
        x_test = normalizer.fit_transform(x_test)
    if scale_flag == True:
        scaler = StandardScaler()
        x_train=scaler.fit_transform(x_train)
        x_test=scaler.transform(x_test)
    if pca_flag == True:
        pca=PCA(n_components=2)
        x_train = pca.fit_transform(x_train)
        x_test = pca.fit_transform(x_test)

    model_class = model_var.fit(x_train,y_train)
    y_hat=model_class.predict(x_test)

    print(f"Accuracy Score {modelName}: {accuracy_score(y_test,y_hat)}")
```

```
    print(f"Confusion Matrix  {modelName}: {confusion_matrix(y_test,y_hat)}")
    print(f"Balanced Accuracy {modelName}:␣
↪{balanced_accuracy_score(y_test,y_hat)}")
    print(f"Classification_Report {modelName}:␣
↪{classification_report(y_test,y_hat)}")



    print(f"-------------------------------------------------------")
    plot_confusion_matrix(model_var,x_test,y_test)
```

Applying SVD Algorithm

[ ]:
```
df_svd= svd_data[['uid', 'id', 'sentiment']].copy()
del svd_data
```

[ ]:
```
encode=ce.
↪OrdinalEncoder(cols=['sentiment'],handle_missing='return_nan',return_df=␣
↪True)
df_svd=encode.fit_transform(df_svd)
```

[ ]:
```
df_svd.sentiment.value_counts()
```

[ ]:
```
1.0    673232
2.0    131330
3.0     15426
Name: sentiment, dtype: int64
```

[ ]:
```
reader=Reader(rating_scale=(1, 3))
data = Dataset.load_from_df(df_svd[['uid', 'id', 'sentiment']], reader)
```

[ ]:
```
svd_train,svd_test=train_test_split(data,test_size=.20)
model = SVD()
model.fit(svd_train)
```

[ ]:
```
<surprise.prediction_algorithms.matrix_factorization.SVD at 0x247cf8f4fa0>
```

[ ]:
```
pred_test=model.test(svd_test)
accuracy.rmse(pred_test)
```

```
RMSE: 0.0831
```

[ ]:
```
0.08312481239006464
```

[ ]:
```
pred=model.predict(15,10)
pred.est
```

[ ]:
```
1
```

[ ]:
```
cross_validate(model,data,cv=5,verbose=True)
```

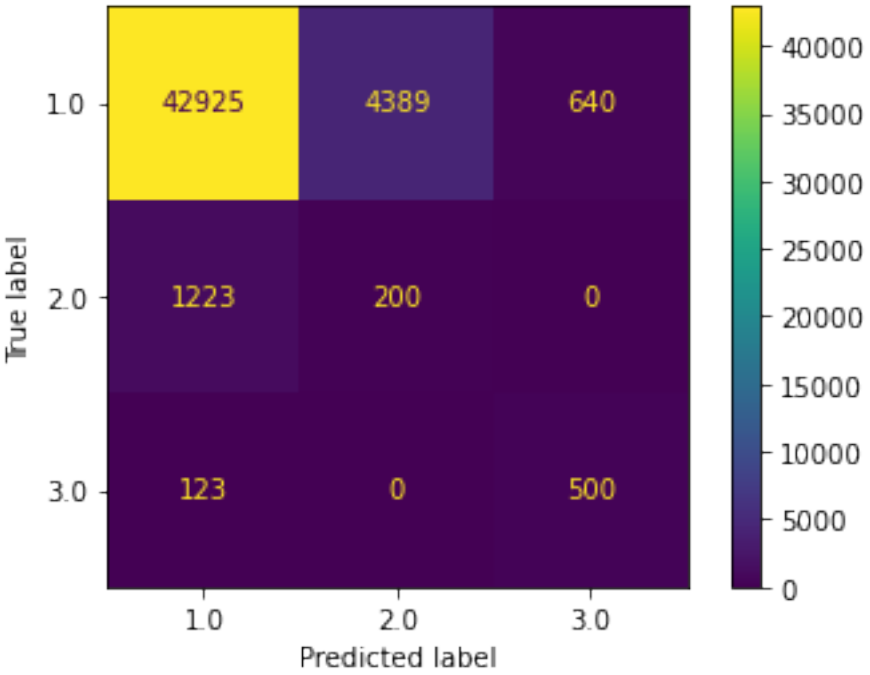
```
Evaluating RMSE, MAE of algorithm SVD on 5 split(s).

                  Fold 1  Fold 2  Fold 3  Fold 4  Fold 5  Mean    Std
RMSE (testset)    0.0838  0.0853  0.0852  0.0840  0.0849  0.0846  0.0006
MAE (testset)     0.0354  0.0359  0.0360  0.0354  0.0358  0.0357  0.0002
Fit time          28.24   28.60   28.46   28.47   28.47   28.45   0.12
Test time         1.28    1.05    1.08    1.28    1.06    1.15    0.11
```

[ ]:
```
{'test_rmse': array([0.08380559, 0.08527449, 0.08518199, 0.08401503,
0.08487453]),
 'test_mae': array([0.03542969, 0.03594819, 0.03599999, 0.03543335,
0.03583647]),
 'fit_time': (28.238356113433838,
  28.597787141799927,
  28.458404302597046,
  28.470713138580322,
  28.465381860733032),
 'test_time': (1.2832889556884766,
  1.0502362251281738,
  1.0782546997070312,
  1.2783007621765137,
  1.0552377700805664)}
```

Apply Other Models

[ ]:
```
smote_flag=True
norm_flag=False
scale_flag=True
pca_flag=False
decision_tree=tree.DecisionTreeClassifier()
check_standalone_prediction(smote_flag,norm_flag,scale_flag,pca_flag,x_train.
 ↪copy(deep=False),y_train.copy(deep=False),x_test.copy(deep=False),y_test.
 ↪copy(deep=False),decision_tree,"Decision Tree")
```

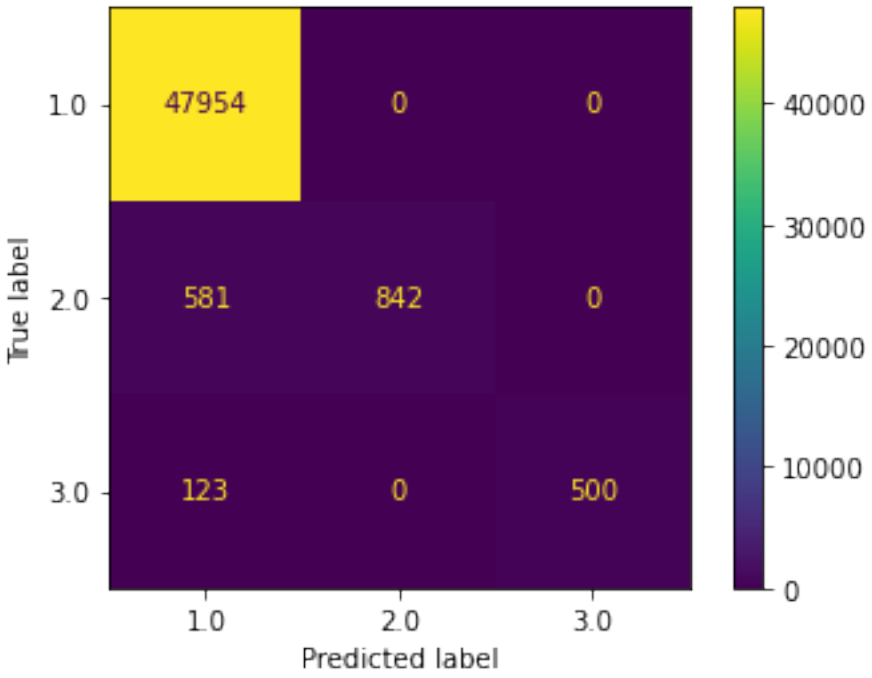
```
Accuracy Score Decision Tree: 0.8725
Confusion Matrix  Decision Tree: [[42925  4389   640]
 [ 1223   200     0]
 [  123     0   500]]
Balanced Accuracy Decision Tree: 0.6127483403354425
Classification_Report Decision Tree:               precision    recall  f1-score
support

         1.0       0.97      0.90      0.93     47954
         2.0       0.04      0.14      0.07      1423
         3.0       0.44      0.80      0.57       623

    accuracy                           0.87     50000
   macro avg       0.48      0.61      0.52     50000
weighted avg       0.94      0.87      0.90     50000
```



[ ]:
```
smote_flag=True
norm_flag=False
scale_flag=True
pca_flag=False
random_state_class=RandomForestClassifier()
check_standalone_prediction(smote_flag,norm_flag,scale_flag,pca_flag,x_train.
↪copy(deep=False),y_train.copy(deep=False),x_test.copy(deep=False),y_test.
↪copy(deep=False),random_state_class,"Random Forest Tree")
```

```
<ipython-input-44-29664ce375fa>:18: DataConversionWarning: A column-vector y was
passed when a 1d array was expected. Please change the shape of y to
(n_samples,), for example using ravel().
  model_class = model_var.fit(x_train,y_train)

Accuracy Score Random Forest Tree: 0.98592
Confusion Matrix  Random Forest Tree: [[47954     0     0]
 [  581   842     0]
 [  123     0   500]]
Balanced Accuracy Random Forest Tree: 0.7980919593906873
Classification_Report Random Forest Tree:               precision    recall
f1-score   support

         1.0       0.99      1.00      0.99     47954
```

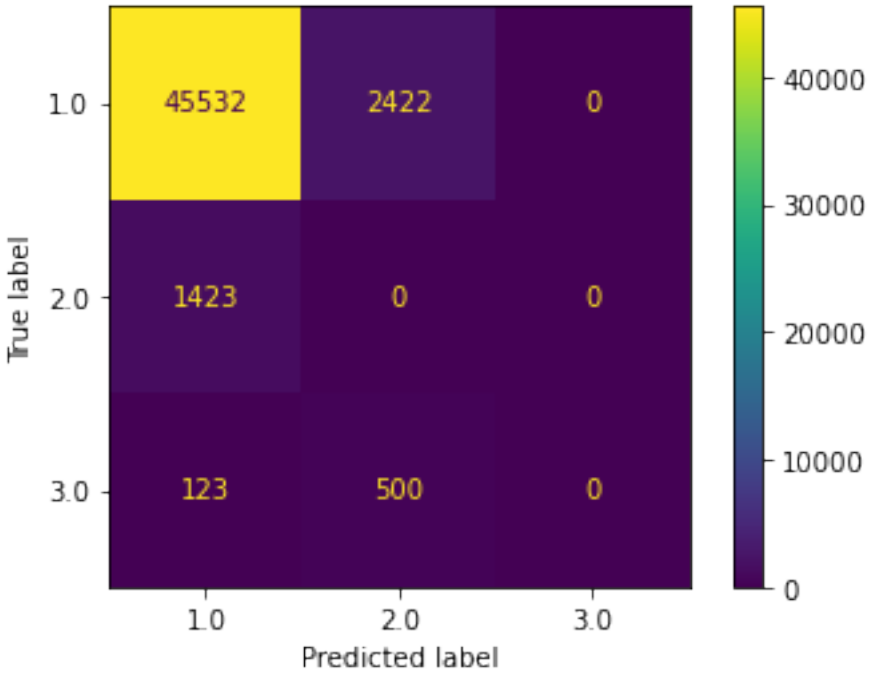
```
           2.0       1.00      0.59      0.74      1423
           3.0       1.00      0.80      0.89       623

      accuracy                           0.99     50000
     macro avg       1.00      0.80      0.88     50000
  weighted avg       0.99      0.99      0.98     50000

--------------------------------------------------------
```



[ ]:
```
smote_flag=True
norm_flag=False
scale_flag=False
pca_flag=True
knnclassifer = KNeighborsClassifier(n_neighbors=2)
check_standalone_prediction(smote_flag,norm_flag,scale_flag,pca_flag,x_train.
 ↪copy(deep=False),y_train.copy(deep=False),x_test.copy(deep=False),y_test.
 ↪copy(deep=False),knnclassifer,"Knn Model")
```

```
C:\Users\pmogh\anaconda3\lib\site-
packages\sklearn\neighbors\_classification.py:179: DataConversionWarning: A
column-vector y was passed when a 1d array was expected. Please change the shape
of y to (n_samples,), for example using ravel().
  return self._fit(X, y)

Accuracy Score Knn Model: 0.91064
Confusion Matrix  Knn Model: [[45532  2422     0]
```

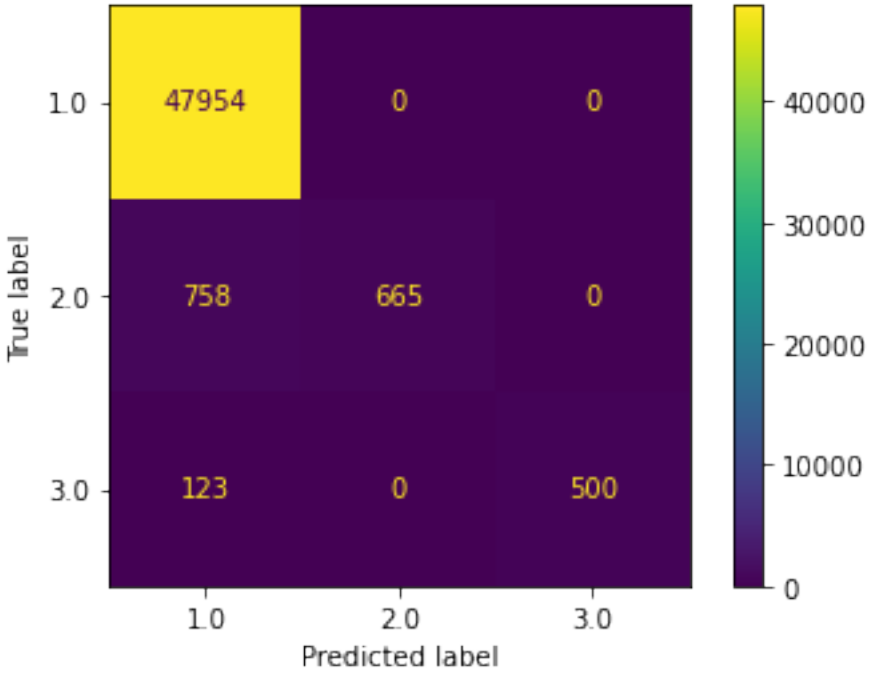
```
 [ 1423     0     0]
 [  123   500     0]]
Balanced Accuracy Knn Model: 0.3164977547927875
Classification_Report Knn Model:               precision    recall  f1-score
support

         1.0       0.97      0.95      0.96     47954
         2.0       0.00      0.00      0.00      1423
         3.0       0.00      0.00      0.00       623

    accuracy                           0.91     50000
   macro avg       0.32      0.32      0.32     50000
weighted avg       0.93      0.91      0.92     50000

------------------------------------------------------

C:\Users\pmogh\anaconda3\lib\site-
packages\sklearn\metrics\_classification.py:1245: UndefinedMetricWarning:
Precision and F-score are ill-defined and being set to 0.0 in labels with no
predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\pmogh\anaconda3\lib\site-
packages\sklearn\metrics\_classification.py:1245: UndefinedMetricWarning:
Precision and F-score are ill-defined and being set to 0.0 in labels with no
predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\pmogh\anaconda3\lib\site-
packages\sklearn\metrics\_classification.py:1245: UndefinedMetricWarning:
Precision and F-score are ill-defined and being set to 0.0 in labels with no
predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```



[ ]:
```
smote_flag=True
norm_flag=False
scale_flag=True
pca_flag=False
svm_class=svm.SVC()
check_standalone_prediction(smote_flag,norm_flag,scale_flag,pca_flag,x_train.
 ↪copy(deep=False),y_train.copy(deep=False),x_test.copy(deep=False),y_test.
 ↪copy(deep=False),svm_class,"SVM Model Tree")
```

```
C:\Users\pmogh\anaconda3\lib\site-packages\sklearn\utils\validation.py:63:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  return f(*args, **kwargs)

Accuracy Score SVM Model Tree: 0.98238
Confusion Matrix  SVM Model Tree: [[47954     0     0]
 [  758   665     0]
 [  123     0   500]]
Balanced Accuracy SVM Model Tree: 0.7566302587582209
Classification_Report SVM Model Tree:               precision    recall
f1-score   support

         1.0       0.98      1.00      0.99     47954
         2.0       1.00      0.47      0.64      1423
         3.0       1.00      0.80      0.89       623
```

```
    accuracy                           0.98     50000
   macro avg       0.99      0.76      0.84     50000
weighted avg       0.98      0.98      0.98     50000

--------------------------------------------------------
```



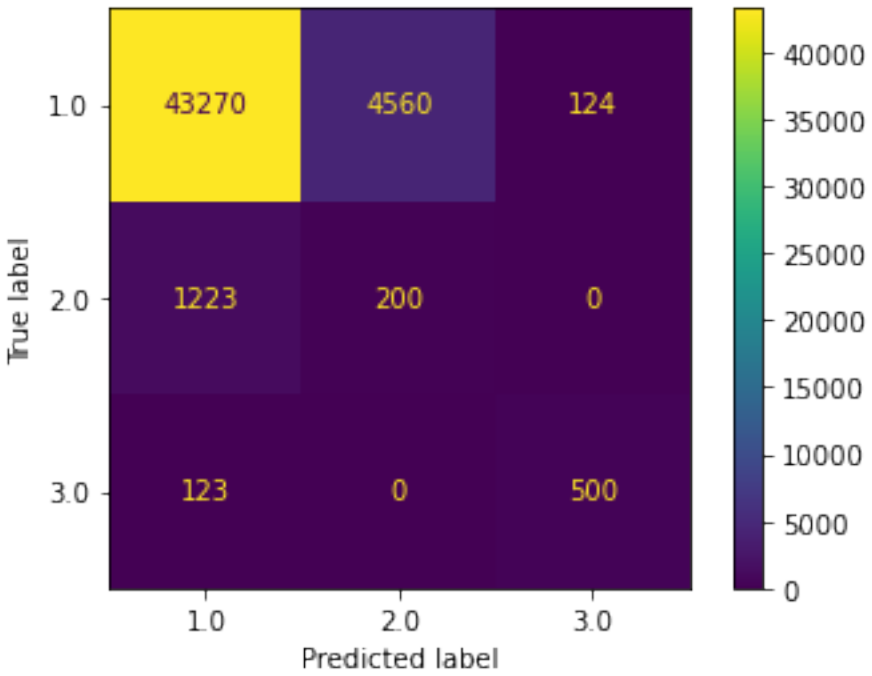
[ ]:
```python
def hyper_parameter(pipe,param,x_train,y_train,x_test,y_test,model_name):
    rsv=GridSearchCV(pipe,param,cv=10,n_jobs=-1)
    rsv.fit(x_train, y_train)
    score = rsv.score(x_test, y_test)
    y_hat=rsv.predict(x_test)
    print("Score:",score)
    print("Best Param",rsv.best_params_)
    print("Best estimator",rsv.best_estimator_)
    print(f"Accuracy Score {model_name}: {accuracy_score(y_test,y_hat)}")
    print(f"Confusion Matrix  {model_name}: {confusion_matrix(y_test,y_hat)}")
    print(f"Balanced Accuracy {model_name}:␣
↪{balanced_accuracy_score(y_test,y_hat)}")
    print(f"Classification_Report {model_name}:␣
↪{classification_report(y_test,y_hat)}")

    print(f"--------------------------------------------------------")
    plot_confusion_matrix(rsv,x_test,y_test)
```

[ ]:
```
def hyper_parameter_random(pipe,param,x_train,y_train,x_test,y_test,model_name):
    rsv=RandomizedSearchCV(pipe,param,cv=10,n_jobs=-1)
    rsv.fit(x_train, y_train)
    score = rsv.score(x_test, y_test)
    y_hat=rsv.predict(x_test)
    print("Score:",score)
    print("Best Param",rsv.best_params_)
    print("Best estimator",rsv.best_estimator_)
    print(f"Accuracy Score {model_name}: {accuracy_score(y_test,y_hat)}")
    print(f"Confusion Matrix  {model_name}: {confusion_matrix(y_test,y_hat)}")
    print(f"Balanced Accuracy {model_name}:␣
 ↪{balanced_accuracy_score(y_test,y_hat)}")
    print(f"Classification_Report {model_name}:␣
 ↪{classification_report(y_test,y_hat)}")

    print(f"-------------------------------------------------------")
    plot_confusion_matrix(rsv,x_test,y_test)
```

[ ]:
```
## Create Copy Of The Test Data
x_train_copy=x_train.copy(deep=False)
y_train_copy=y_train.copy(deep=False)
x_test_copy=x_test.copy(deep=False)
y_test_copy=y_test.copy(deep=False)
## Apply Smote On The Data
sm=SMOTE(random_state=42)
x_train_copy, y_train_copy = sm.fit_resample(x_train_copy, y_train_copy)
#x_test_copy, y_test_copy = sm.fit_resample(x_test_copy, y_test_copy)


params = {"clf__criterion": ['gini', 'entropy'],"clf__max_depth":
 ↪[10,30,50,100,None]}

pipe = Pipeline(steps=[('transformer',StandardScaler()),("clf", tree.
 ↪DecisionTreeClassifier())])
#pca 86
#without pca 99
hyper_parameter(pipe,params,x_train_copy,y_train_copy,x_test_copy,y_test_copy,"Decision␣
 ↪Tree")
del x_train_copy
del y_train_copy
del x_test_copy
del y_test_copy

del params
del pipe
```

```
Score: 0.8794
```

```
Best Param {'clf__criterion': 'gini', 'clf__max_depth': 10}
Best estimator Pipeline(steps=[('transformer', StandardScaler()),
                ('clf', DecisionTreeClassifier(max_depth=10))])
Accuracy Score Decision Tree: 0.8794
Confusion Matrix  Decision Tree: [[43270  4560   124]
 [ 1223   200     0]
 [  123     0   500]]
Balanced Accuracy Decision Tree: 0.6151464718781708
Classification_Report Decision Tree:               precision    recall  f1-score
support

         1.0       0.97      0.90      0.93     47954
         2.0       0.04      0.14      0.06      1423
         3.0       0.80      0.80      0.80       623

    accuracy                           0.88     50000
   macro avg       0.60      0.62      0.60     50000
weighted avg       0.94      0.88      0.91     50000

----------------------------------------------------------
```



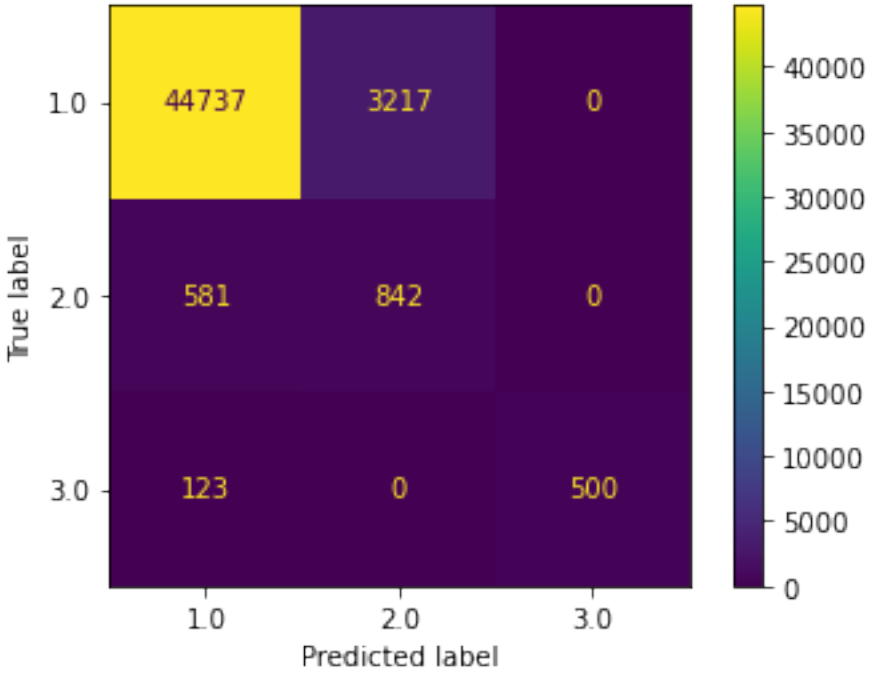
[ ]:
```
## Create Copy Of The Test Data
x_train_copy=x_train.copy(deep=False)
y_train_copy=y_train.copy(deep=False)
x_test_copy=x_test.copy(deep=False)
```

```
y_test_copy=y_test.copy(deep=False)
## Apply Smote On The Data
sm=SMOTE(random_state=42)
x_train_copy, y_train_copy = sm.fit_resample(x_train_copy, y_train_copy)
#x_test_copy, y_test_copy = sm.fit_resample(x_test_copy, y_test_copy)

params = {"clf__n_estimators": [10,30,50,100],"clf__max_depth":␣
 ↪[10,50,None],"clf__max_features": [5, 10],"clf__bootstrap":
 ↪[True,False],"clf__criterion":['gini','entropy']}

pipe = Pipeline(steps=[("transformer", StandardScaler()),("clf",␣
 ↪RandomForestClassifier())])

hyper_parameter_random(pipe,params,x_train_copy,y_train_copy,x_test_copy,y_test_copy,"Random␣
 ↪Forest")
del x_train_copy
del y_train_copy
del x_test_copy
del y_test_copy

del params
del pipe
```

```
C:\Users\pmogh\anaconda3\lib\site-packages\sklearn\pipeline.py:346:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples,), for example using
ravel().
  self._final_estimator.fit(Xt, y, **fit_params_last_step)
```

```
Score: 0.92158
Best Param {'clf__n_estimators': 10, 'clf__max_features': 10, 'clf__max_depth':
10, 'clf__criterion': 'entropy', 'clf__bootstrap': False}
Best estimator Pipeline(steps=[('transformer', StandardScaler()),
                ('clf',
                 RandomForestClassifier(bootstrap=False, criterion='entropy',
                                        max_depth=10, max_features=10,
                                        n_estimators=10))])
Accuracy Score Random Forest: 0.92158
Confusion Matrix  Random Forest: [[44737  3217     0]
 [  581   842     0]
 [  123     0   500]]
Balanced Accuracy Random Forest: 0.7757302516429846
Classification_Report Random Forest:               precision    recall  f1-score
support

         1.0       0.98      0.93      0.96     47954
         2.0       0.21      0.59      0.31      1423
         3.0       1.00      0.80      0.89       623
```

```
    accuracy                           0.92     50000
   macro avg       0.73      0.78      0.72     50000
weighted avg       0.96      0.92      0.94     50000

--------------------------------------------------------
```



Part-5-Recommend_Model

December 21, 2021

1 Recommendation Techniques and Results

In this file, using merged data of game and player we develop the working recommendation system

[ ]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

from scipy import sparse

from lightfm import LightFM
from sklearn.metrics.pairwise import cosine_similarity
from lightfm.evaluation import precision_at_k
from lightfm.evaluation import auc_score

from scipy.spatial import distance

from sklearn.manifold import TSNE
from gensim.models.keyedvectors import Word2VecKeyedVectors
import textwrap

from support_function import *

#Import Warnings
import warnings
warnings.filterwarnings("ignore")
```

Load the supported function from support_function.py file in this notebook.

[ ]:
```
%run support_function.py
```

1.1 Get the Data For recommendation from cvs file.

We load the csv file which contain the userId and game id in it.

[ ]:
```
user_recom_data = pd.read_csv('recdata.csv', index_col=0)
user_recom_data = user_recom_data.rename(columns = {'variable':'id', 'value':␣
↪'owned'})
```

```
user_recom_data.head()
```

[ ]:
```
   uid  id  owned
0    0  10    1.0
1    1  10    1.0
2    3  10    1.0
3    4  10    1.0
4   10  10    1.0
```

We load the games data with names and all other information like geners

[ ]:
```
# Load games data
game_recom_data = pd.read_csv('gamesdata.csv', index_col = 0)
game_recom_data.head()
```

[ ]:
```
          publisher                                              genres  \
0         Kotoshiro  ['Action', 'Casual', 'Indie', 'Simulation', 'S…
1  Making Fun, Inc.       ['Free to Play', 'Indie', 'RPG', 'Strategy']
2      Poolians.com  ['Casual', 'Free to Play', 'Indie', 'Simulatio…
3                                  ['Action', 'Adventure', 'Casual']
4               NaN                                                NaN

                  app_name                    title  \
0      Lost Summoner Kitty      Lost Summoner Kitty
1                Ironbound                Ironbound
2  Real Pool 3D - Poolians  Real Pool 3D - Poolians
3                     2222                     2222
4            Log Challenge                      NaN

                                           url release_date  \
0  http://store.steampowered.com/app/761140/Lost_…   2018-01-04
1  http://store.steampowered.com/app/643980/Ironb…   2018-01-04
2  http://store.steampowered.com/app/670290/Real_…   2017-07-24
3     http://store.steampowered.com/app/767400/2222/   2017-12-07
4  http://store.steampowered.com/app/773570/Log_C…          NaN

                                           tags  discount_price  \
0  ['Strategy', 'Action', 'Indie', 'Casual', 'Sim…            4.49
1  ['Free to Play', 'Strategy', 'Indie', 'RPG', '…             NaN
2  ['Free to Play', 'Simulation', 'Sports', 'Casu…             NaN
3                  ['Action', 'Adventure', 'Casual']            0.83
4            ['Action', 'Indie', 'Casual', 'Sports']            1.79

                                         reviews_url  \
0  http://steamcommunity.com/app/761140/reviews/?…
1  http://steamcommunity.com/app/643980/reviews/?…
2  http://steamcommunity.com/app/670290/reviews/?…
3  http://steamcommunity.com/app/767400/reviews/?…
```

```
4  http://steamcommunity.com/app/773570/reviews/?…

                                                specs         price  \
0                                  ['Single-player']          4.99
1  ['Single-player', 'Multi-player', 'Online Mult…  Free To Play
2  ['Single-player', 'Multi-player', 'Online Mult…  Free to Play
3                                  ['Single-player']          0.99
4  ['Single-player', 'Full controller support', '…          2.99

   early_access        id         developer        sentiment  metascore
0         False  761140.0         Kotoshiro              NaN        NaN
1         False  643980.0  Secret Level SRL  Mostly Positive        NaN
2         False  670290.0      Poolians.com  Mostly Positive        NaN
3         False  767400.0                               NaN        NaN
4         False  773570.0               NaN              NaN        NaN
```

1.2 Additional Preprocessing

1.2.1 Create interaction matrix

We will create an interactions matrix using the user-item data. This is done using the `create_interaction_matrix` function, which can be found in `support_function.py`.

[ ]:
```
inter_mat = build_matrix(data_frame= user_recom_data,
                                        u_column = 'uid',
                                        i_column = 'id',
                                        r_column = 'owned')
inter_mat.shape
```

[ ]:
```
(8769, 8171)
```

From the shape, we note that we have 8769 unique users and 8171 different games represented.

[ ]:
```
# Preview head
inter_mat.head(10)
```

[ ]:
```
id    10      20      30      40      50      60      70      80      130     \
uid
0       1.0     1.0     1.0     1.0     1.0     1.0     1.0     0.0     1.0
1       1.0     1.0     1.0     1.0     1.0     1.0     1.0     1.0     1.0
2       0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
3       1.0     1.0     1.0     1.0     1.0     1.0     1.0     1.0     1.0
4       1.0     1.0     1.0     1.0     1.0     1.0     1.0     1.0     1.0
5       0.0     0.0     0.0     0.0     1.0     0.0     0.0     0.0     0.0
6       0.0     1.0     0.0     0.0     1.0     0.0     1.0     0.0     1.0
7       0.0     0.0     0.0     0.0     0.0     0.0     1.0     0.0     0.0
8       0.0     1.0     0.0     0.0     1.0     0.0     1.0     0.0     1.0
10      1.0     1.0     1.0     1.0     1.0     1.0     1.0     1.0     1.0
```

```
id    220     …  525190  526460  526790  527340  527440  527510  527520  \
uid           …
0       1.0   …     0.0     0.0     0.0     0.0     0.0     0.0     0.0
1       1.0   …     0.0     0.0     0.0     0.0     0.0     0.0     0.0
2       1.0   …     0.0     0.0     0.0     0.0     0.0     0.0     0.0
3       1.0   …     0.0     0.0     0.0     0.0     0.0     0.0     0.0
4       1.0   …     0.0     0.0     0.0     0.0     0.0     0.0     0.0
5       1.0   …     0.0     0.0     0.0     0.0     0.0     0.0     0.0
6       1.0   …     0.0     0.0     0.0     0.0     0.0     0.0     0.0
7       1.0   …     0.0     0.0     0.0     0.0     0.0     0.0     0.0
8       1.0   …     0.0     0.0     0.0     0.0     0.0     0.0     0.0
10      1.0   …     0.0     0.0     0.0     0.0     0.0     0.0     0.0

id    527810  527900  530720
uid
0       0.0     0.0     0.0
1       0.0     0.0     0.0
2       0.0     0.0     0.0
3       0.0     0.0     0.0
4       0.0     0.0     0.0
5       0.0     0.0     0.0
6       0.0     0.0     0.0
7       0.0     0.0     0.0
8       0.0     0.0     0.0
10      0.0     0.0     0.0

[10 rows x 8171 columns]
```

1.2.2 Break Down the dataset into train and test

[ ]:
```
# Get number of users
len(inter_mat)
```

[ ]:
```
8769
```

We choose to have roughly 80% of our data as training and 20% as test.

[ ]:
```
# Establish number of users in train/test sets

train_num = round((80/100)*len(inter_mat),0)
print(f'We desire {train_num} users in our training set.')

test_num = len(inter_mat)-train_num
print(f'We desire {test_num} users in our test set.')
```

```
We desire 7015.0 users in our training set.
We desire 1754.0 users in our test set.
```

[ ]:
```
# Define train and test sets
train = inter_mat[:7015]
test = inter_mat[7015:]
```

1.2.3 Build Player Dictionary

[ ]:
```
# Create user dictionary using helper function
player_dic = build_dic_user(matrix_inter=inter_mat)
```

1.2.4 Build Game Dictionary

[ ]:
```
# Create game dictionary using helper function
dic_game = build_dic_item(data_frame=game_recom_data, column_id= 'id',␣
 ↪column_n='title')
```

1.2.5 Build Sparse Matrix

We will transform the interaction into a sparse matrix, to make computations efficient.

For the trainset, we simply use the `sparse.csr_matrix()` function.

With the test set, due to a known issue, we need to add additional rows so that the number of rows matches the trains set.

[ ]:
```
# Create sparse matrices for evaluation
train_sparse = sparse.csr_matrix(train.values)

#Add X users to Test so that the number of rows in Train match Test
N = train.shape[0] #Rows in Train set
n,m = test.shape #Rows & columns in Test set
z = np.zeros([(N-n),m]) #Create the necessary rows of zeros with m columns
#test = test.toarray() #Temporarily convert Test into a numpy array
test = np.vstack((test,z)) #Vertically stack Test on top of the blank users
test_sparse = sparse.csr_matrix(test) #Convert back to sparse
```

1.3 Using Hybrid Model LightFM for the Recommendation

1.3.1 Input WARP loss model

[ ]:
```
mf_model_warp = apply_model(matrix_inter= train,
                  component_val= 30,
                  loss = 'warp',
                  epoch = 30,
                  n_jobs = 4)
```

[ ]:
```
train_precision = precision_at_k(mf_model_warp, train_sparse, k=10).mean()
test_precision = precision_at_k(mf_model_warp, test_sparse, k=10).mean()
print('Precision: train %.2f, test %.2f.' % (train_precision, test_precision))
```

```
Precision: train 0.73, test 0.43.
```

[ ]:
```
train_auc = auc_score(mf_model_warp, train_sparse).mean()
test_auc = auc_score(mf_model_warp, test_sparse).mean()
print('AUC: train %.2f, test %.2f.' % (train_auc, test_auc))
```

```
AUC: train 0.98, test 0.92.
```

AUC score is very good on both train and test

1.3.2 Input BPR loss model

[ ]:
```
mf_model_bpr = apply_model(matrix_inter= train,
                  component_val= 30,
                  loss = 'bpr',
                  epoch = 30,
                  n_jobs = 4)
```

[ ]:
```
train_precision = precision_at_k(mf_model_bpr, train_sparse, k=10).mean()
test_precision = precision_at_k(mf_model_bpr, test_sparse, k=10).mean()
print('Precision: train %.2f, test %.2f.' % (train_precision, test_precision))
```

```
Precision: train 0.80, test 0.36.
```

[ ]:
```
train_auc = auc_score(mf_model_bpr, train_sparse).mean()
test_auc = auc_score(mf_model_bpr, test_sparse).mean()
print('AUC: train %.2f, test %.2f.' % (train_auc, test_auc))
```

```
AUC: train 0.96, test 0.71.
```

Again, the AUC score is good, though significantly lower for the test set compare to the train set.

Based on these two models, we will keep WARP as the loss function due to better performance all round.

1.3.3 Hyper Tuning of the model by varing components

The `n_components` parameter controlls the number of embeddings (dimension of the features in the latent space.)

We will vary this number, lowering it to `10` first and then increasing it to `50` to see how this affects model performance.

[ ]:
```
mf_model_warp_2 = apply_model(matrix_inter= train,
                  component_val= 10,
                  loss = 'warp',
                  epoch = 30,
                  n_jobs = 4)
```

[ ]:
```
train_precision = precision_at_k(mf_model_warp_2, train_sparse, k=10).mean()
test_precision = precision_at_k(mf_model_warp_2, test_sparse, k=10).mean()
```

```
print('Precision: train %.2f, test %.2f.' % (train_precision, test_precision))
```

```
Precision: train 0.68, test 0.48.
```

[ ]:
```
train_auc = auc_score(mf_model_warp_2, train_sparse).mean()
test_auc = auc_score(mf_model_warp_2, test_sparse).mean()
print('AUC: train %.2f, test %.2f.' % (train_auc, test_auc))
```

```
AUC: train 0.97, test 0.93.
```

[ ]:
```
mf_model_warp_50 = apply_model(matrix_inter= train,
                  component_val= 50,
                  loss = 'warp',
                  epoch = 30,
                  n_jobs = 4)
```

[ ]:
```
train_precision = precision_at_k(mf_model_warp_50, train_sparse, k=10).mean()
test_precision = precision_at_k(mf_model_warp_50, test_sparse, k=10).mean()
print('Precision: train %.2f, test %.2f.' % (train_precision, test_precision))
```

```
Precision: train 0.76, test 0.42.
```

[ ]:
```
train_auc = auc_score(mf_model_warp_50, train_sparse).mean()
test_auc = auc_score(mf_model_warp_50, test_sparse).mean()
print('AUC: train %.2f, test %.2f.' % (train_auc, test_auc))
```

```
AUC: train 0.99, test 0.91.
```

In the end performance is not change on the base of component

1.3.4 Build Model

[ ]:
```
# Instantiate and fit model on full interactions set
light_fm_mode = apply_model(matrix_inter= inter_mat,
                  component_val = 30,
                  loss = 'warp',
                  epoch = 30,
                  n_jobs = 4)
```

1.4 Embeddings

Apply Embedding Space of the Model

1.4.1 Retrieve embeddings matrix

[ ]:
```
# Get embeddings
embeddings = light_fm_mode.item_embeddings
embeddings
```

[ ]:
```
array([[-0.07690857,  0.66486156,  0.7936622 , …,  0.14951605,
        -0.0695671 , -0.6245182 ],
       [-0.60291576,  0.83658946,  0.58764905, …,  0.5114703 ,
        -0.3604954 , -1.0809734 ],
       [-0.2310924 ,  1.0992005 ,  0.64228845, …,  0.98967004,
         0.06214582, -0.49858683],
       …,
       [ 0.64778465, -0.10613031,  0.09715862, …,  0.57438576,
        -0.2034148 , -0.07921824],
       [ 0.27167612,  0.5803558 , -0.41966903, …,  0.6099575 ,
         0.5862158 , -0.12392855],
       [ 0.35757646,  0.32205895, -0.3167676 , …,  0.53355885,
         0.01019462,  0.25211778]], dtype=float32)
```

[ ]:
```
embeddings.shape
```

[ ]:
```
(8171, 30)
```

Check the Game Vector

[ ]:
```
embeddings[0]
```

[ ]:
```
array([-0.07690857,  0.66486156,  0.7936622 ,  0.40528223, -0.145565  ,
       -0.36309764, -0.34237215,  0.15373814,  0.54323393, -0.5631929 ,
       -0.61242783, -0.29522717,  0.5400072 ,  0.34600753, -0.74698734,
       -0.2089371 ,  0.31638432, -0.23969008, -0.64567953, -0.39736956,
        0.1493338 ,  0.21329583,  0.7203192 , -0.6489853 , -0.0677067 ,
       -0.31437442,  0.27613798,  0.14951605, -0.0695671 , -0.6245182 ],
      dtype=float32)
```

Find the game name from the matrix

[ ]:
```
firstgameid = inter_mat.columns[0]
dic_game[firstgameid]
```

[ ]:
```
'Counter-Strike'
```

1.4.2 Pair similarity

Let find the distance between the similar game

We search the 'Counter-Strike' game

[ ]:
```
game_recom_data[(game_recom_data['title']=='Counter-Strike') |␣
↪(game_recom_data['title']=='Left 4 Dead 2') ]
```

[ ]:
```
      publisher      genres        app_name           title  \
31529     Valve  ['Action']   Left 4 Dead 2   Left 4 Dead 2
32106     Valve  ['Action']  Counter-Strike  Counter-Strike
```

```
                                                    url release_date  \
31529  http://store.steampowered.com/app/550/Left_4_D…   2009-11-16
32106  http://store.steampowered.com/app/10/CounterSt…   2000-11-01

                                                    tags  discount_price  \
31529  ['Zombies', 'Co-op', 'FPS', 'Multiplayer', 'Ac…             NaN
32106  ['Action', 'FPS', 'Multiplayer', 'Shooter', 'C…             NaN

                                              reviews_url  \
31529  http://steamcommunity.com/app/550/reviews/?bro…
32106  http://steamcommunity.com/app/10/reviews/?brow…

                                                  specs  price  early_access  \
31529  ['Single-player', 'Multi-player', 'Co-op', 'St…  19.99         False
32106       ['Multi-player', 'Valve Anti-Cheat enabled']   9.99         False

          id developer                sentiment  metascore
31529  550.0     Valve  Overwhelmingly Positive       89.0
32106   10.0     Valve  Overwhelmingly Positive       88.0
```

Check the Vector of the games

[ ]:
```
cs_index = 0
cs_vector = embeddings[cs_index]
cs_vector
```

[ ]:
```
array([-0.07690857,  0.66486156,  0.7936622 ,  0.40528223, -0.145565  ,
       -0.36309764, -0.34237215,  0.15373814,  0.54323393, -0.5631929 ,
       -0.61242783, -0.29522717,  0.5400072 ,  0.34600753, -0.74698734,
       -0.2089371 ,  0.31638432, -0.23969008, -0.64567953, -0.39736956,
        0.1493338 ,  0.21329583,  0.7203192 , -0.6489853 , -0.0677067 ,
       -0.31437442,  0.27613798,  0.14951605, -0.0695671 , -0.6245182 ],
      dtype=float32)
```

[ ]:
```
lfd2_id = game_recom_data[game_recom_data['title']=='Left 4 Dead 2']['id'].
 ↪values[0]
lfd2_index = list(inter_mat.columns).index(lfd2_id)
lfd2_vector = embeddings[lfd2_index]
lfd2_vector
```

[ ]:
```
array([-0.01273242, -0.07405546,  0.6101665 ,  0.1412955 , -0.330248  ,
       -0.9404025 , -0.29786164, -0.25623107,  0.19409557,  0.40056136,
       -0.60454905,  0.01597847,  0.29765967,  1.0109664 , -0.6728011 ,
        0.5028252 , -0.14399414, -0.2512675 , -0.32039464, -0.43069765,
        0.16283731,  0.60680556,  0.1759727 , -0.2787006 , -0.5939248 ,
        0.5459706 ,  0.28189695, -0.43314165, -0.60124415, -0.62766606],
      dtype=float32)
```

To assign a single value to the similarity between these two vectors, we calculate the distance between them. Let us first compute the Euclidean distance.

[ ]:
```
# Compute Euclidean distance
distance.euclidean(cs_vector, lfd2_vector)
```

[ ]:
```
2.4286489486694336
```

Let us compare this figure with a pair of games we believe to be very different.

[ ]:
```
# Get data for both games
game_recom_data[(game_recom_data['title']=='Counter-Strike') |␣
↪(game_recom_data['title']=='The Room') ]
```

[ ]:
```
            publisher                  genres        app_name  \
2472   Fireproof Games  ['Adventure', 'Indie']        The Room
32106            Valve              ['Action']  Counter-Strike

               title                                                 url  \
2472         The Room  http://store.steampowered.com/app/288160/The_R…
32106  Counter-Strike  http://store.steampowered.com/app/10/CounterSt…

      release_date                                               tags  \
2472    2014-07-28  ['Puzzle', 'Adventure', 'Point & Click', 'Indi…
32106   2000-11-01  ['Action', 'FPS', 'Multiplayer', 'Shooter', 'C…

       discount_price                                     reviews_url  \
2472              NaN  http://steamcommunity.com/app/288160/reviews/?…
32106             NaN  http://steamcommunity.com/app/10/reviews/?brow…

                                                specs price  early_access  \
2472   ['Single-player', 'Steam Achievements', 'Steam…  4.99         False
32106       ['Multi-player', 'Valve Anti-Cheat enabled']  9.99         False

             id        developer                 sentiment  metascore
2472   288160.0  Fireproof Games  Overwhelmingly Positive       73.0
32106      10.0            Valve  Overwhelmingly Positive       88.0
```

[ ]:
```
# Retrieve game id for The Room
room_id = game_recom_data[game_recom_data['title']=='The Room']['id'].values[0]

# Obtain index for Squad in interactions matrix
room_index = list(inter_mat.columns).index(room_id)

# Obtain embeddings vector
room_vector = embeddings[room_index]

room_vector
```

[ ]:
```
array([-0.1024242 ,  0.4752257 ,  0.24124612, -0.00741749,  0.21419159,
       -0.28054076, -0.03180434, -0.38254467,  0.13352183, -0.33322603,
        0.13092154, -0.15414314,  0.00489135, -0.19807154, -0.33829376,
       -0.07191804, -0.17468366,  0.18996401,  0.08875996,  0.30820638,
        0.35630605,  0.00255273, -0.5786068 ,  0.22534879,  0.0135782 ,
       -0.45574978, -0.27468544, -0.31314114, -0.21707602, -0.04487913],
      dtype=float32)
```

[ ]:
```
# Compute Euclidean distance
distance.euclidean(cs_vector, room_vector)
```

[ ]:
```
2.738154172897339
```

Check the distance with cosine distance

[ ]:
```
print(f'Cosine distance between Counter Strike and Left 4 Dead 2: {distance.
 ↪cosine(cs_vector, lfd2_vector)}')
print(f'Cosine distance between Counter Strike and The Room: {distance.
 ↪cosine(cs_vector, room_vector)}')
```

```
Cosine distance between Counter Strike and Left 4 Dead 2: 0.4738824963569641
Cosine distance between Counter Strike and The Room: 0.9319570809602737
```

1.4.3 Exploring embeddings with Gensim

[ ]:
```
embedding_size = embeddings.shape[1]
kv = Word2VecKeyedVectors(embedding_size)

gameslist = []
for game_id in inter_mat.columns:
    name = dic_game[game_id]
    gameslist.append(name)

kv.add_vectors(gameslist, embeddings)
```

Let us obtain the games closest to `Counter-Strike`.

[ ]:
```
kv.most_similar('Counter-Strike')
```

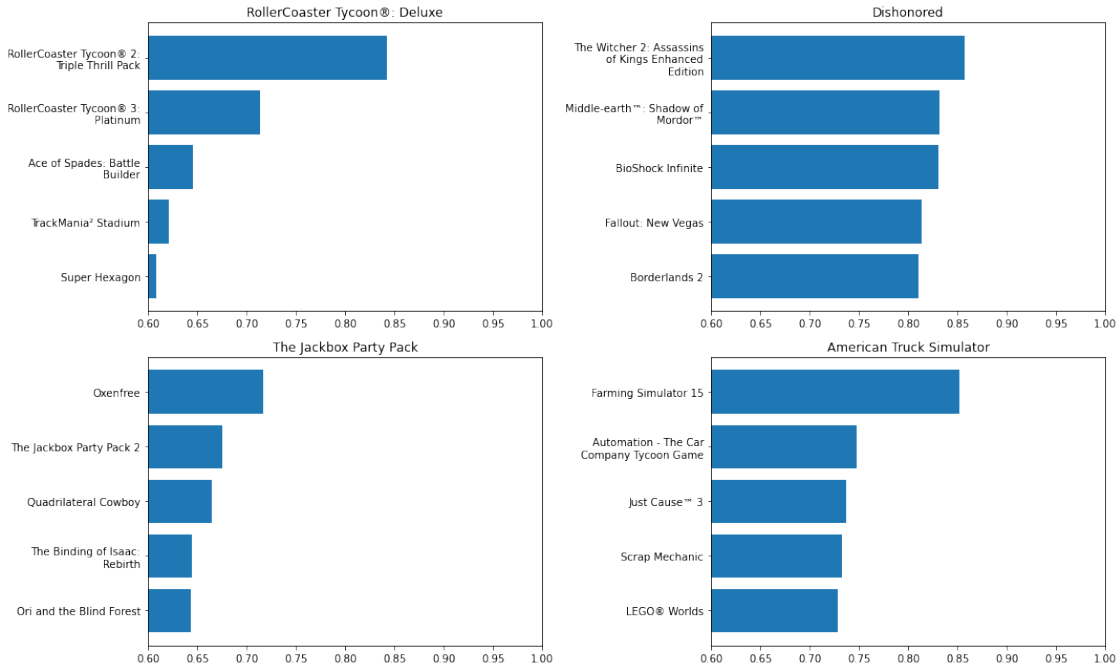
[ ]:
```
[('Counter-Strike: Condition Zero', 0.9755569100379944),
 ('Day of Defeat', 0.8177651762962341),
 ('Half-Life: Source', 0.8058030009269714),
 ('Day of Defeat: Source', 0.8011133074760437),
 ('Team Fortress Classic', 0.7981202006340027),
 ('Half-Life: Blue Shift', 0.7888646721839905),
 ('Half-Life', 0.7878614068031311),
 ('Ricochet', 0.7845215797424316),
 ('Deathmatch Classic', 0.7804338335990906),
 ('Half-Life: Opposing Force', 0.7775993347167969)]
```

[ ]:
```
kv.most_similar('Left 4 Dead 2')
```

[ ]:
```
[('PAYDAY  The Heist', 0.9239243865013123),
 ('Borderlands 2', 0.8823978900909424),
 ("Garry's Mod", 0.8753825426101685),
 ('Counter-Strike: Global Offensive', 0.8716615438461304),
 ('Killing Floor', 0.8585674166679382),
 ('Defence Alliance 2', 0.8505985140800476),
 ('The Ship: Murder Party', 0.842189610004425),
 ('Saints Row IV', 0.8398149609565735),
 ('Terraria', 0.8383126854896545),
 ('Chivalry: Medieval Warfare', 0.8292756676673889)]
```

[ ]:
```
def build_graph(game, x, best_num=5):
    sim = kv.most_similar(game, topn=best_num)[::-1]
    y = np.arange(len(sim))
    w = [t[1] for t in sim]
    x.barh(y, w)
    left = min(.6, min(w))
    x.set_xlim(right=1.0, left=left)
    # Split long titles over multiple lines
    labels = [textwrap.fill(t[0] , width=24)
              for t in sim]
    x.set_yticks(y)
    x.set_yticklabels(labels)
    x.set_title(game)
```

[ ]:
```
list_game = ['RollerCoaster Tycoon®: Deluxe', 'Dishonored',
         'The Jackbox Party Pack', 'American Truck Simulator']

figure, ax = plt.subplots(2, 2, figsize=(15, 9))

for game, ax in zip(list_game, ax.flatten()):
    build_graph(game, ax)

figure.tight_layout()
```



1.5 User Recommendations

1.5.1 Recommendations for existing user

[ ]:
```
# Get recommendations
rec_list_u12 = find_recom(rec_mode= light_fm_mode,
                    matrix_inter= inter_mat,
                    player_id= 5000,
                    dic_player= player_dic,
                    dic_game= dic_game,
                    threshold = 0,
                    num_items = 5,
                    show_known = True,
                    show_recs = True)
```

```
Games Likes:
1- Dungeon Fighter Online
2- ESEA
3- H1Z1 Test Server
4- H1Z1
5- CS:GO Player Profiles
6- Just Survive Test Server
7- AdVenture Capitalist
8- Strife®
9- Dirty Bomb®
10- Game of Thrones - A Telltale Games Series
```

11- Don't Starve Together
12- Unturned
13- Robocraft
14- Mount Your Friends
15- Warface
16- Fistful of Frags
17- The Walking Dead: Season 2
18- Rust
19- Toribash
20- Heroes & Generals
21- FEZ
22- No More Room in Hell
23- Cry of Fear
24- Insurgency
25- Hotline Miami
26- MapleStory
27- The Walking Dead
28- Castle Crashers®
29- Max Payne 3
30- Mount & Blade: Warband
31- Super Meat Boy
32- Defence Alliance 2
33- TrackMania Nations Forever
34- BioShock Infinite
35- Just Cause 2
36- Garry's Mod
37- Killing Floor
38- Counter-Strike: Global Offensive
39- Left 4 Dead 2
40- Left 4 Dead
41- Half-Life 2: Episode Two
42- Portal
43- Half-Life 2: Episode One
44- Half-Life Deathmatch: Source
45- Half-Life 2: Lost Coast
46- Half-Life 2: Deathmatch
47- Counter-Strike: Source
48- Half-Life 2

Game Recommended Items:
1- Portal 2
2- Terraria
3- Rocket League®
4- Grand Theft Auto V
5- Warframe

1.6 Item Recommendations

1.6.1 Create item embedding matrix

[ ]:
```
game_mat_data =  build_embedding_mat(rec_mode=light_fm_mode,␣
↪matrix_inter=inter_mat)
```

[ ]:
```
game_mat_data.shape
```

[ ]:
```
(8171, 8171)
```

[ ]:
```
game_mat_data.head()
```

[ ]:
```
id    10        20        30        40        50        60        70      \
id
10  1.000000  0.798120  0.817765  0.780434  0.777599  0.784522  0.787861
20  0.798120  1.000000  0.867307  0.852161  0.994024  0.866077  0.977754
30  0.817765  0.867307  1.000000  0.990534  0.862059  0.990114  0.843498
40  0.780434  0.852161  0.990534  1.000000  0.849804  0.994848  0.820985
50  0.777599  0.994024  0.862059  0.849804  1.000000  0.862647  0.981975

id    80        130       220     …    525190    526460    526790    527340  \
id                                  …
10  0.975557  0.788865  0.720709  … -0.306754 -0.330438 -0.341211 -0.054234
20  0.699381  0.995552  0.846169  … -0.293373 -0.205190 -0.240755 -0.133135
30  0.719655  0.871238  0.718637  … -0.162556 -0.096175 -0.150723 -0.181699
40  0.670797  0.858557  0.672284  … -0.095033 -0.022256 -0.077084 -0.152839
50  0.675560  0.996402  0.849154  … -0.301745 -0.195885 -0.240017 -0.145447

id    527440    527510    527520    527810    527900    530720
id
10 -0.361544 -0.485556 -0.328589 -0.066259 -0.281063 -0.482814
20 -0.242896 -0.352428 -0.174543 -0.130799 -0.206378 -0.334659
30 -0.069702 -0.258437 -0.061448  0.054523 -0.009513 -0.191741
40 -0.026975 -0.190437  0.011434  0.098432  0.068500 -0.126358
50 -0.224202 -0.325862 -0.143163 -0.115081 -0.192277 -0.309017

[5 rows x 8171 columns]
```

1.6.2 Generate item recommendations

[ ]:
```
item_rec_list_10 = return_recomm( embedd_mat= game_mat_data,
                                  game_id= 10,
                                  dic_game= dic_game,
                                  n_items = 6,
                                  show = True)
```

```
Item of interest: Counter-Strike
Similar items:
```

```
1- Counter-Strike: Condition Zero
2- Day of Defeat
3- Half-Life: Source
4- Day of Defeat: Source
5- Team Fortress Classic
6- Half-Life: Blue Shift
```

[ ]:
```
game_recom_data[game_recom_data['title'] == 'The Witness']
```

[ ]:
```
         publisher                   genres     app_name        title  \
5211  Thekla, Inc.  ['Adventure', 'Indie']  The Witness  The Witness

                                                  url release_date  \
5211  http://store.steampowered.com/app/210970/The_W…   2016-01-26

                                                 tags  discount_price  \
5211  ['Puzzle', 'Exploration', 'First-Person', 'Sin…             NaN

                                           reviews_url  \
5211  http://steamcommunity.com/app/210970/reviews/?…

                                                specs  price  early_access  \
5211  ['Single-player', 'Steam Achievements', 'Capti…  39.99         False

            id     developer      sentiment  metascore
5211  210970.0  Thekla, Inc.  Very Positive       87.0
```

[ ]:
```
game_210970 = return_recomm( embedd_mat= game_mat_data,
                             game_id = 210970,
                             dic_game= dic_game,
                             n_items = 5,
                             show = True)
```

```
Item of interest: The Witness
Similar items:
1- Bear Simulator
2- Everybody's Gone to the Rapture
3- OPUS: The Day We Found Earth
4- ABZU
5- Dangerous Golf
```

[ ]:
```
game_recom_data[game_recom_data['title'] == 'ABZU']
```

[ ]:
```
       publisher                                              genres app_name  \
22123  505 Games  ['Action', 'Adventure', 'Casual', 'Indie', 'Si…     ABZU

      title                                          url release_date  \
22123  ABZU  http://store.steampowered.com/app/384190/ABZU/   2016-08-02
```

```
                                                      tags  discount_price  \
22123  ['Relaxing', 'Atmospheric', 'Underwater', 'Gre…             NaN

                                              reviews_url  \
22123  http://steamcommunity.com/app/384190/reviews/?…

                                                   specs  price  early_access  \
22123  ['Single-player', 'Steam Achievements', 'Full …  19.99         False

             id    developer      sentiment  metascore
22123  384190.0  Giant Squid  Very Positive       83.0
```

[ ]:
```
game_384190 = return_recomm( embedd_mat= game_mat_data,
                             game_id = 384190,
                             dic_game= dic_game,
                             n_items = 6,
                             show = True)
```

```
Item of interest: ABZU
Similar items:
1- Quadrilateral Cowboy
2- The Witness
3- Valley
4- Replica
5- 1979 Revolution: Black Friday
6- Everybody's Gone to the Rapture
```